Joint Information Extraction and Compression of Satellite Image Time Series

Lionel Gueguen

▶ **To cite this version:**

Lionel Gueguen. Joint Information Extraction and Compression of Satellite Image Time Series. domain_other. Télécom ParisTech, 2007. English. NNT : . pastel-00003146

HAL Id: pastel-00003146
https://pastel.hal.science/pastel-00003146

Submitted on 16 Jan 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TELECOM PARIS
école nationale supérieure des télécommunications

École Doctorale d'Informatique, Télécommunications et Électronique de Paris

Thèse

Présentée pour obtenir le grade de docteur
de l'École Nationale Supérieure des Télécommunications

Spécialité : **Signal et Images**

Lionel Gueguen

Sujet :

EXTRACTION D'INFORMATION ET COMPRESSION CONJOINTES DES SÉRIES TEMPORELLES D'IMAGES SATELLITAIRES

SOUTENUE LE 30 OCTOBRE 2007 DEVANT LE JURY COMPOSÉ DE :

M.	BRUZZONE LORENZO	Président
M.	MOHAMMAD-DJAFARI ALI	Rapporteur
M.	MARTHON PHILIPPE	Rapporteur
Mme.	LAMBERT CATHERINE	Examinateur
M.	MAITRE HENRI	Examinateur
M.	DATCU MIHAI	Directeur de thèse

Table des matières

Remerciements

Je remercie avant tout Mihai Datcu pour avoir assuré la direction de ma thèse avec enthousiasme, disponibilité et rigueur scientifique. Je remercie aussi Alain Giros et Henri Maitre pour leurs conseils avisés. Qu'ils trouvent ici l'expression de ma gratitude pour leurs conseils et pour la confiance qu'ils m'ont accordée tout au long de mes recherches.

Je remercie également les membres du jury de m'avoir fait l'honneur de participer à ma soutenance : Lorenzo Bruzzone, président du jury, Ali Mohammad-Djafari et Philippe Marthon, rapporteur, Catherine Lambert, examinatrice.

Je remercie toutes les personnes du département TSI : les enseignants-chercheurs, les secrétaires, les responsables informatiques, les doctorants, les post-doctorants et les stagiaires. En particulier, je remercie les personnes avec qui j'ai passé des moments inoubliables, que ce soit à la pause café ou au zinc d'un troquet de la Butte aux Cailles ; Christophe, Greg, Maria, Dora, Yole qui m'ont chaleureusement intégré à mon début de thèse ; Mathieu, Lois, Cléo, Miguel, Fabrice, Laurence, Nancy, Jean, Cyril, Slim, Valentin, Aurélia, Sarah, Pierre avec qui j'ai partagé de trés bons moments de rigolade au FIAP et ailleurs ; Tyz' qui est le meilleur co-box ; Seb et Jo qui m'ont fait souffrir au sport.

Je tiens à remercier Fabio, Riton, Loic, Garsaint, Guillaume, Eiffel, Baps, Pierre, Tanguy qui m'ont supporté moralement pendant ces trois années.

Je remercie mes amis de Brest, en particulier Pierre-Yves, Ronan, Aurélie et Aurélien, pour leur soutien et leur amitié tout au long de ces années. Je remercie et félicite mes parents et ma soeur pour m'avoir permis d'accomplir cette thèse. Finalement, je tiens à remercier ma compagne Suzanne pour son soutien inconditionnel pendant cette dernière année de thèse qui fût très prenante.

Introduction

Le premier système permettant de photographier la Terre depuis l'espace fût embarqué sur le satellite Explorer-6 lancé en 1959 par la NASA. Depuis les satellites pour l'observation de la Terre n'ont cessé de se développer et de s'améliorer pour offrir une vision globale de plus en plus détaillée de la surface terrestre. Ces observations visent à mieux appréhender et comprendre notre environnement. Par exemple, les satellites météorologiques nous renseignent sur le climat, les radars embarqués permettent la construction de modèles en 3-dimensions du globe, le satellite GRACE a permis de dresser une carte de la gravité terrestre et les satellites optiques apportent des images haute résolution de la couverture terrestre. Et cette diversification dans l'analyse de la Terre continue toujours sa progression.
Avec le nombre croissant de capteurs optiques en orbite et la grande agilité des satellites, il est désormais possible d'observer régulièrement une même scène. Ainsi, ce que nous appèllerons les Séries Temporelles d'Images Satellitaires (STIS) peut être construit au segment sol en agrégeant les différentes acquisitions temporellement espacées. Ces séries s'apparentent aux observations météorologiques puisqu'elles caractérisent une évolution. Cependant les STIS constituent de nouvelles données pour l'observation de la Terre puisqu'elles bénéficient de la haute résolution spatiale apportée par les capteurs mis en jeu. Plus que l'observation spatiale, les STIS permettent d'observer spatio-temporellement la surface du globe.
Le nombre d'images satellitaires est en augmentation constante. Pour les exploiter pleinement, des outils dédiés au traitement automatique du contenu informationnel sont développés. Ces outils sont une aide précieuse pour les utilisateurs qui peuvent mieux cibler l'information qui les intéresse et mieux l'analyser. Par exemple, nous trouvons des outils de recherche fondés sur le contenu informationnel, des outils de segmentation et des outils de classification. Cependant les STIS confèrent une nouvelle dimension à l'observation de la Terre sur deux points. Premièrement, l'évolution temporelle des images ne permet pas la réutilisation des outils dédiés à l'image. Deuxièmement, les STIS sont des données qui s'accroissent et qui engrangent de nouvelles informations au fur et à mesure des acquisitions. Cet accroissement permanent des données peut poser à long terme des problèmes de stockage. Enfin, l'exploitation des Séries Temporelles d'Images Satellitaires constitue un enjeu majeur pour un nombre grandissant de domaines d'application intéressés par la compréhension de l'évolution de la couverture terrestre.
L'objectif de cette thèse est de fournir de nouvelles méthodologies pour analyser et compresser conjointement les nombreux événements spatio-temporels qui apparaissent dans les STIS. L'objectif est de créer une représentation des STIS qui soit compacte et qui décrive le contenu informationnel avec parcimonie. Cette représentation doit être universelle, c'est-à-dire qu'elle pourra être utilisée pour des applications dédiées à la compréhension telles que la fouille d'information, l'étiquetage automatique des structures spatio-

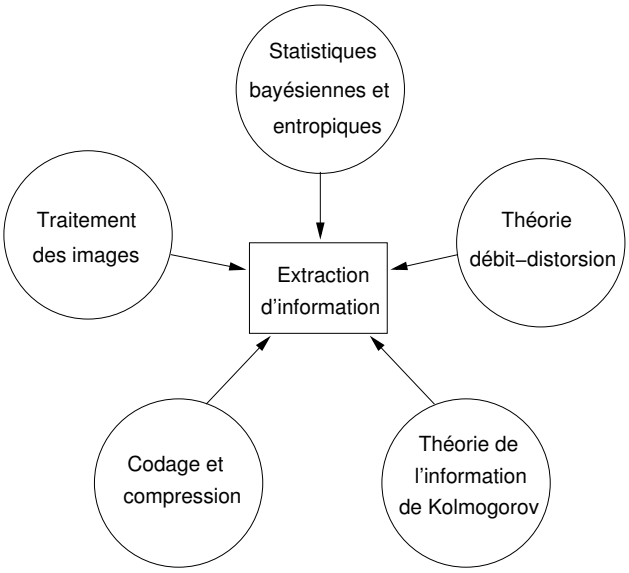


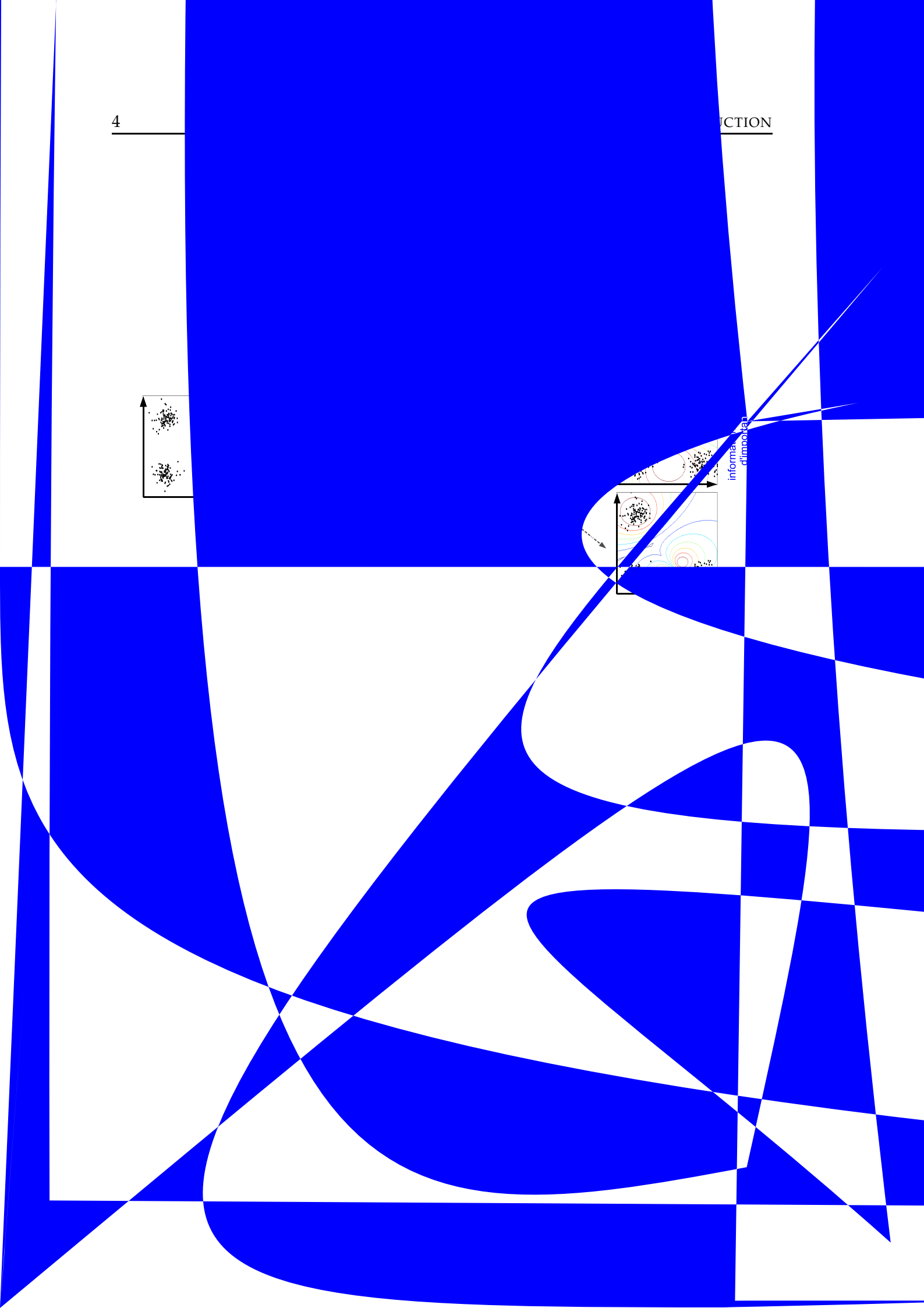
FIG. 1 – L'extraction d'information à la rencontre de deux domaines (compression et traitement d'images) et de trois courants de pensée (Bayes, Shannon et Kolmogorov).

temporelles, la découverte de nouveaux événements et la recherche par le contenu.
Pour accomplir cet objectif, nous nous plaçons dans le cadre de l'extraction d'information à partir des données brutes. Ce domaine en plein essor avec le boom de l'information proliférant sur la toile est un cadre tout à fait adéquat pour innover dans la représentation du contenu informationnel. L'extraction d'information consiste à reproduire une partie de l'information totale véhiculée par les données qui représente un intérêt particulier pour une application spécifique. Dans le cadre de la thèse nous serons attentifs à ce que l'information extraite soit la plus représentative du contenu informationnel véhiculé par les phénomènes spatio-temporels.
Les techniques et méthodes d'extraction d'information sont foisonnantes. Dans le cadre de la thèse, nous restreignons l'extraction d'information au croisement de deux domaines et trois courants de pensée (cf. Figure 1). Les deux domaines mis en jeux sont le traitement d'images et la compression puisque nous souhaitons obtenir une représentation compacte de séries d'images. D'autre part, concernant la vision de l'extraction d'information, nous nous appuyons sur les statistiques bayésiennes et entropiques, la théorie débit-distorsion due à Shannon et la théorie de l'information de Kolmogorov. Ces trois courants de pensée ont des implications directes dans le domaine du codage et de la compression.
Ce manuscrit est segmenté en trois parties :

- la première traite de l'intégration de l'extraction d'information et de la compression dans les STIS.
- la deuxième traite de nos contributions pour le développement de deux nouvelles méthodologies pour l'extraction d'information des STIS.
- la troisième traite de la validation des méthodologies et de l'analyse des résultats obtenus sur les STIS.

La partie I donne une vision approfondie du contexte de la thèse en décrivant la problématique, l'état de l'art de la fouille d'information, le concept de l'intégration de la fouille et de la compression et enfin le cadre théorique du concept. Le chapitre 1 présente la STIS

particulière sur laquelle notre travail s'est porté. Le chapitre 2 présente la problématique de la fouille d'information. Nous y présentons le paradigme de la fouille d'information et établissons les limites dans lesquelles intervient l'extraction d'information. En effet, nous présentons un nouveau concept pour intégrer l'indexation informationnelle à la compression d'une base d'objets (Stoian, 1988). Ensuite, nous présentons l'état de l'art des techniques d'extraction d'information des images, des vidéos, des STIS et de médias usuels compressés. Enfin, nous décrivons les liens qui unissent les techniques d'extraction d'information et les techniques de compression qui nous permettront d'introduire le nouveau concept d'intégration de l'extraction et de la compression. Finalement, le chapitre 3 décrit le cadre théorique pour l'extraction et la compression. Dans ce chapitre, nous décrivons les trois courants de pensée liés à Bayes, Shannon et Kolmogorov. En particulier, nous rappelons les passerelles qui relient ces trois mouvements et nous mettons en évidence les apports de ces théories pour unifier l'extraction d'information et la compression.
La partie II regroupe nos contributions méthodologiques pour l'extraction d'information des STIS. Le chapitre 4 présente une méthodologie inspirée de la théorie débit-distorsion et fondée sur le principe d'*Information Bottleneck* (cf. Figure 2) pour à la fois extraire la quantité optimale d'information d'importance et donner une compression avec pertes des événements spatio-temporels qui composent la STIS. Pour construire cette méthodologie, nous avons d'abord établi des liens avec le principe de Longueur de Description Minimale (LDM) qui nous a permis de cibler l'information d'intérêt. Ensuite, nous donnons un critère heuristique sur les courbes débit-distorsion pour extraire les quantités d'information optimales. Enfin, pour appliquer cette méthodologie générale aux STIS, nous nous sommes appuyés sur des modèles stochastiques tels que les champs aléatoires de Gibbs-Markov pour modéliser les évolutions des textures. Il résulte de cette méthode une indexation des structures spatio-temporelles, qui est une représentation compacte et parcimonieuse du contenu informationnel, et dont la quantité d'information véhiculée est contrôlée. D'autre part, nous avons déduit le critère de *Multi-Information Bottleneck* et l'algorithme qui en découle. Ce principe permet de pondérer l'importance de différents types d'information indépendants. Nous l'appliquons aux STIS en considérant l'information de couleur et d'évolution des textures. Enfin, nous finissons par décrire un nouveau principe dual du principe d'*Information Bottleneck*. Nous l'appelons *Variational Information Bottleneck* et nous donnons les équations consistantes associées. Le chapitre 5 présente une seconde méthodologie pour extraire l'information et donner une compression sans perte d'objets informationnels (cf. Figure 3). Nous construisons cette méthodologie générale dans le cadre de la théorie de l'information de Kolmogorov. Cette méthodologie, dont certains liens avec le principe d'*Information Bottleneck* sont dressés, permet d'intégrer une indexation du contenu informationnel dans le code. La technique déduite de la méthodologie permet de remplir complètement nos objectifs en produisant une représentation compacte (le code) qui inclut une description du contenu informationnel par un index. Dans le but de construire la technique d'indexation et codage conjoint, nous introduisons une nouvelle distance informationnelle fondée sur les longueurs de code. Cette méthode générale est appliquée aux STIS par l'introduction d'un compresseur sans perte en deux parties construit spécialement pour ces séries temporelles.
Enfin, la partie III donne une validation et une analyse des résultats avant de conclure et donner les perspectives. Dans le chapitre 6, nous validons les trois méthodologies en les appliquant à des données synthétiques dont nous connaissons le contenu informationnel. Ensuite, nous donnons une analyse visuelle des résultats obtenus sur la STIS présentée


informa
d'import

au chapitre 1. Et nous soulignons l'intérêt particulier de la méthode du chapitre 5 pour l'extraction d'information et la compression en y attachant un système de recherche par exemple. Ce système permet de fouiller dans le code avec très peu de décodage. Finalement la conclusion présente une synthèse des travaux dont nous dégageons des perspectives pour la poursuite de ces travaux sur un plan théorique et applicatif.

Première partie

Intégration de l'extraction d'information et de la compression des Séries Temporelles d'Images Satellitaires : problématique, concept et théorie

Chapitre 1

Les Séries Temporelles d'Images Satellitaires (STIS)

Nous décrivons dans ce chapitre les Séries Temporelles d'Images Satellitaires. Ces données sont constituées d'images optiques acquises par les Satellites Pour l'Observation de la Terre (SPOT). Elles constituent une partie de la base du projet d'Assimilation de Données par Agro Modélisation (ADAM[1]). Ce projet a comme buts principaux l'évaluation et l'apport des techniques d'assimilation de données spatiales dans le domaine des modèles agronomiques, en vue de développer des applications répondant aux besoins en information agricole.

1.1 Caractérisation des données

La STIS, sur laquelle nous travaillons, relate la dynamique d'une zone rurale du département d'Ilfov en Roumanie, à l'Est de Bucarest (cf. Figures 1.1) durant les années 2000-2002. Les satellites SPOT possèdent des radiomètres embarqués permettant d'acquérir des images optiques de la Terre. Il y a 3 générations de capteurs embarqués dans les satellites. La première génération regroupe les satellites SPOT-1,2 et 3, lancés respectivement en 1986, 1990 et 1993. SPOT-3 est le seul à ne plus être en fonctionnement à présent. La seconde génération est arrivée avec SPOT-4 lancé en 1998. Enfin, la troisième génération à haute résolution spatiale est apparue avec le lancement de SPOT-5 en 2002. Dans le cadre du projet ADAM, les images optiques ont été acquises par SPOT-1,2 et 4. Chaque image est composée de 3 bandes spectrales du domaine des ondes visibles et proches du visible. La première bande est dans le "vert-jaune", la deuxième est dans le "rouge" et la troisième est dans le "proche infra-rouge". Ces trois bandes correspondent respectivement aux intervalles de longueurs d'ondes allant de 0.50 à 0.59 μm, de 0.61 à 0.68 μm et de 0.79 à 0.89 μm. Les satellites SPOT sont en orbite quasi polaire, circulaire, héliosynchrone à une altitude de 822 km. Ainsi, ils peuvent acquérir des images numériques ligne par ligne du fait de leur mouvement relatif par rapport à la Terre. Les lignes ont une longueur de 3000 pixels. Leurs capteurs ont une haute résolution spatiale, puisqu'un pixel représente environ un carré de taille 20×20 m^2 à la surface du globe. En réalité, les pixels ont une longueur de 20 m et une largeur variant de 20 m à 27 m à cause de l'inclinaison. Mais les images sont échantillonnées pour obtenir une résolution

[1]Les données STIS sont en accès libre sur le site http ://kalideos.cnes.fr/, moyennant votre inscription gratuite au site. Images SPOT : copyright CNES, 2000-2003

FIG. 1.1 – Carte de Roumanie et des alentours de Bucarest présentant la scène qui a été imagée à plusieurs reprises pour constituer la STIS. Cette zone se situe dans le département d'Iflov à l'Est de Bucarest. Crédit photo http ://maps.google.com

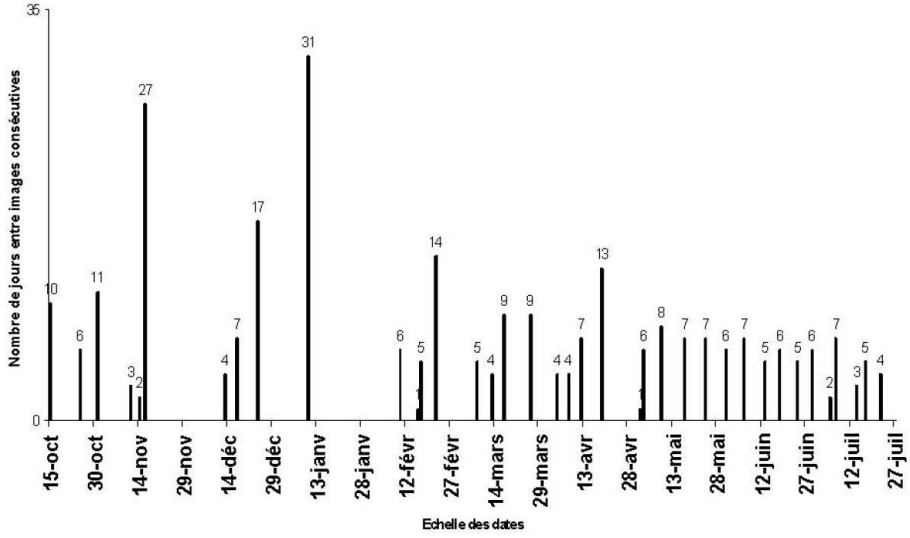


FIG. 1.2 – Nombre de jours entre images consécutives de la STIS finale. L'axe horizontal présente l'échantillonnage irrégulier en fonction des dates régulières. L'axe vertical représente la différence de temps entre deux dates consécutives.

de 20×20 m^2.
L'acquisition d'images de la même scène rurale a été journalière durant une période de 10 mois, allant d'octobre 2000 à juillet 2001. Pour construire la STIS ADAM, les images ont ensuite subi plusieurs étapes de sélection et de prétraitement. Nous détaillons par la suite les différents processus postérieurs à l'acquisition pour la constitution du jeu de données.

1.2 Constitution de la STIS ADAM

1.2.1 Sélection des images

Dans le cadre du projet ADAM, pour mieux répondre aux besoins de l'agro-assimilation, les images acquises quotidiennement ont été sélectionnées en supprimant les images présentant une couverture neigeuse ou nuageuse sur des sites de culture préalablement identifiés. La STIS finale est donc composée de 38 images irrégulièrement échantillonnées dans le temps sur une période de 286 jours. L'échantillonnage irrégulier est présenté dans la Figure 1.2. Nous remarquons que l'échantillonnage est très irrégulier pendant la saison hivernale avec des écarts de temps allant jusqu'à 31 jours. Pendant la saison estivale, l'échantillonnage est beaucoup plus régulier avec des intervalles de temps n'excédant pas 13 jours. Remarquons que même si la sélection s'est opérée pour supprimer la couverture neigeuse ou nuageuse, certaines images contiennent des nuages ou de la neige en quantité acceptable.

1.2.2 Alignement géométrique

Afin de les aligner dans un même repère cartographique, les 38 images acquises par les satellites sont soumises à des prétraitements géométriques. Un modèle numérique de terrain du site est calculé à partir de deux images stéréoscopiques dans un premier

temps. Les coordonnées géographiques d'une trentaine de points de liaison, visibles dans chacune des 38 images, sont mises en relation avec le modèle numérique de terrain. Ensuite par l'intermédiaire d'une spatio-triangulation et d'interpolations, les images sont projetées dans un même repère géographique puis cartographique. Les images finales obtenues ont une taille de 3000×2000 pixels^2 ; soit une superficie au sol de 60×40 km^2. Par ce traitement, les images finales sont rendues géométriquement superposables. Un pixel représentant un site spatialement localisé sur le globe admet comme zone de variation dans la STIS un disque de diamètre 1.5 pixel. En d'autres termes, sur toute la séquence, un même site se situera dans un cylindre temporel dont le diamètre spatial est 1.5 pixel. Par conséquent il n'y a pas de mouvement dans cette série, puisque les objets géométriques sont stables spatialement.
Cependant des zones de la STIS sont inexploitables. En effet, les coins de certaines images ne possèdent aucune valeur à cause des rotations durant la correction géométrique. Pour obtenir une STIS consistante où chaque pixel correspond à une mesure radiométrique, nous extrayons une sous séquence dont la dimension spatiale est 2500×1500 et qui n'inclut pas ces parties.

1.2.3 Correction radiométrique

Chaque pixel de la série temporelle est corrigé radiométriquement Dans un premier temps, la radiométrie est corrigée par l'utilisation de modèles physiques. Dans un second temps, la radiométrie est ajustée par l'inter calibration entre couple d'images consécutives. Les radiomètres embarqués dans les satellites SPOT mesurent la luminance de la surface terrestre. La luminance représente l'énergie des radiations réfléchies par la surface pour une gamme de longueurs d'ondes donnée. C'est une acquisition passive puisque les capteurs mesurent les radiations solaires réfléchies. Les mesures de luminances sont converties en mesures de réflectance. Ce processus correspond à normaliser la luminance en calculant le rapport entre l'énergie des radiations réfléchies et l'énergie des radiations incidentes provenant du Soleil. Comme la luminance, la réflectance se calcule pour une gamme d'ondes donnée et prend ses valeurs dans l'intervalle $[0, 1]$.
Par la suite, des corrections atmosphériques sont appliquées aux mesures de réflectance par l'intermédiaire de modèles de transfert radiatif. Ces modèles représentent le transfert des ondes aux travers de l'atmosphère en considérant des données extérieures telles que les concentrations d'eau, d'ozone, d'aérosols et l'angle de visée. Ainsi par traitement inverse, les images résultantes mesurent la réflectance au niveau de la canopée. La STIS obtenue contient des mesures indépendantes de l'angle de visée et des phénomènes atmosphériques.
Enfin les images consécutives sont inter-calibrées. Comme les capteurs utilisés pour l'acquisition ne sont pas forcement les mêmes d'une image à l'autre, une inter-calibration est nécessaire. Afin d'atténuer les transitions radiométriques brusques, les images consécutives sont ajustées linéairement. Pendant cet ajustement, les bandes spectrales sont prises indépendamment. Ces dernières images constituent la STIS qui sera traitée dans nos travaux.

1.2.4 Représentation numérique

Comme nous l'avons signalé, la STIS finale est constituée de 38 images de taille 2500×1500 avec 3 bandes spectrales par image. De plus, les réflectances corrigées sont quan-

tifiées sur 10 bits, de façon que les valeurs entières correspondantes soient incluses dans l'intervalle $[0, 1000]$. Une image est codée avec 3.75 Mb par bande, et avec 11.25 Mb au total. Quand les 38 images sont prises en compte, la STIS est représentée avec 427.5 Mb. En conséquence, la STIS contient potentiellement énormément d'information.
Ces STIS nécessitent des moyens de compression pour limiter l'espace de stockage. L'acquisition d'images est permanente et la série s'enrichit de nouvelles images chaque jour. Par exemple, les données du projet ADAM se sont enrichies de nouvelles acquisitions entre 2002 et 2003. Nous ne les considérons pas dans nos expériences.

1.3 Contenu informationnel des STIS

Nous présentons plusieurs sous-séquences de la STIS tirée du projet ADAM dans la Figure 1.3 qui présentent des événements visibles. Les images de la série temporelle ont subi une correction d'histogrammes dynamique (Heas, 2005). Cette correction artificielle permet d'obtenir une homogénéité visuelle. Les STIS sont des objets complexes contenant des structures spatio-temporelles nombreuses et variées : occlusions par des nuages, moissons de cultures parcellaires, inondations, etc. Aussi l'analyse de ces structures spatio-temporelles est utile pour l'étude et la compréhension de phénomènes dans des domaines variés tels que l'agronomie, l'océanographie, l'urbanisme ou la géologie.
Par exemple, nous trouvons le projet GMES (Global Monitoring for Environment and Security) qui a été créé récemment pour la surveillance des ressources naturelles mondiales. Le but principal de ce programme est d'étudier l'évolution de l'occupation des sols en ciblant les zones soumises à des dégradations naturelles ou d'origine humaine. Ces études seront menées grâce au support d'agences spatiales qui sont en mesure de produire des images ou des séries pour l'observation de la Terre. Ainsi, le projet veut apporter les moyens de surveiller et de gérer les ressources naturelles grâce aux informations extraites de l'observation de la Terre. Par conséquent, il est essentiel de produire des outils qui permettent d'automatiser cette tâche d'extraction d'information et qui puissent s'adapter à des besoins différents.
En conclusion, nous voulons créer de nouveaux moyens pour fouiller l'information fournie sous forme brute dans les STIS. Nous observons que les structures spatio-temporelles constituent le coeur de l'information contenue dans la STIS. Ainsi, nous nous attacherons à modéliser ces structures spatio-temporelles et à en extraire l'information pertinente. En l'occurrence, nous souhaitons construire un système de fouille de données dans les STIS pour répondre à des besoins divers tels que l'identification de zones sinistrées après une inondation ou le suivi de cultures particulières. De plus, comme les séries s'enrichissent tous les jours avec des nouvelles observations, les outils doivent être capables d'absorber et d'analyser ces nouvelles données pour fournir à tout moment les informations les plus récentes.

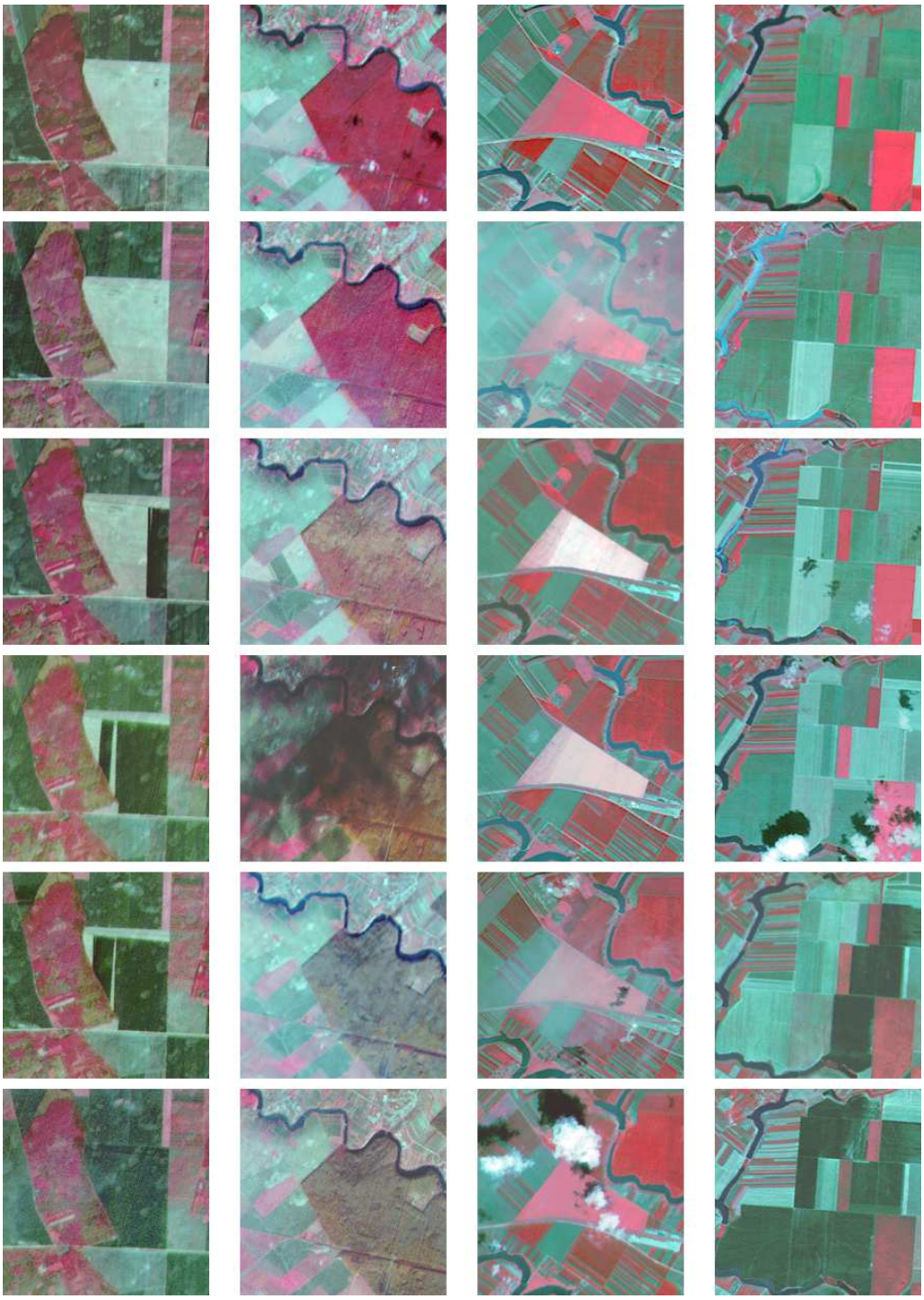


FIG. 1.3 – Nous présentons quatre sous-séquences de la STIS du projet ADAM. Elles se parcourent de haut en bas. Chaque série est composée de 6 images de taille 200×200. Ces sous-séquences présentent quelques phénomènes qu'il est possible d'observer dans la STIS. La première observe la récolte d'un champ qui est l'activité humaine la plus observable du ciel. La seconde présente l'évolution de la forêt à l'automne. Nous pouvons remarquer le passage de l'ombre de la brume dans cette séquence. La troisième la maturation d'un champ de colza qui apparaît en blanc. Finalement, la quatrième série présente des récoltes et une évolution de la rivière. Nous notons des occlusions par des nuages dans les deux séquences de droite.

Chapitre 2

Recherche par le contenu dans les archives de données

Dans ce chapitre, nous donnons l'état de l'art de la recherche d'information par le contenu. Pour commencer, nous abordons les concepts sous-jacents à la recherche par le contenu. Ensuite, nous nous concentrons sur l'extraction d'information dans les images, vidéos et les Séries Temporelles d'Images Satellitaires. Puis, nous décrivons les liens qui existent entre l'extraction d'information et la compression. Ces liens nous permettront d'établir un nouveau concept pour l'extraction non supérvisée d'information.

2.1 La recherche et la fouille d'information

Avec l'émergence des moyens de stockage et de communications numériques, l'information est devenue un produit en plein essor ces dernières années. Avec ce foisonnement il devient de plus en plus difficile d'accéder à l'information pertinente. En outre avoir la bonne information au bon moment est crucial pour prendre la bonne décision dans de nombreux domaines. De nombreuses techniques se sont développées pour rechercher l'information pertinente, telles que les moteurs de recherche par mots clés sur internet. Néanmoins pour certains types d'information, tels que l'image, l'annotation reste encore un moyen répandu pour la recherche. Certes cette méthodologie peut être efficace quand l'annotation est correcte, mais annoter de telles quantités d'information devient humainement impossible. Ainsi, le concept de recherche par le contenu est apparu dans les années 90 pour rechercher automatiquement des images. L'enjeu caché derrière ce concept est d'inférer une sémantique ou du sens à partir d'une image. De nos jours, ce concept d'extraction d'information est appliqué dans bien d'autres applications de recherche par le contenu, comme le diagnostic médical assisté à partir d'images. En effet, les méthodes élaborées permettent de sélectionner et d'extraire l'information suffisante à discriminer ou à retrouver des objets numériques. Cela présuppose que l'on sache évaluer la similitude entre deux informations. Dans le cas où la similitude est calculable, un schéma simple de moteur de recherche par le contenu peut être défini. Un exemple de moteur de recherche par exemple est présenté dans la Figure 2.1. A chaque objet de la base de données sont associées des primitives obtenues au préalable lors d'une étape d'extraction d'information. Ces objets sont ensuite regroupés par similitude afin d'obtenir des ensembles homogènes. Cette étape de groupage est justifiée par une diminution de la complexité des recherches pour d'énormes bases de données. Lors d'une requête, un exemple

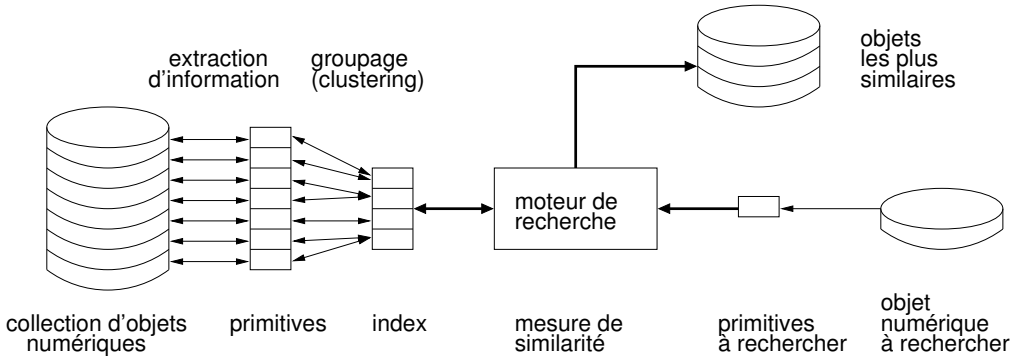


FIG. 2.1 – Architecture d'un moteur de recherche par le contenu. La recherche consiste à retrouver les objets les plus similaires à l'objet passé en exemple.

est donné au moteur de recherche et les primitives de l'objet donné en exemple sont extraites. Ainsi le moteur de recherche, où est définie la mesure de similarité entre informations extraites, peut déterminer l'ensemble d'objets les plus semblables à l'exemple et cet ensemble constitue la réponse à la requête de recherche par exemple. Dans ce schéma, l'extraction d'information, la mesure de similarité et le regroupage sont les points cruciaux pour fonder un système de recherche rapide et présentant de bonnes performances de recherche. Ces dernières années, la nécessité de moteurs de recherche d'images par le contenu s'est accrue pour gérer les collections d'images médicales, satellitaires ou accessibles sur la toile. Ce besoin a ouvert une nouvelle branche dans le domaine du traitement des images dénoté par Recherche d'Image Basée sur le Contenu (RIBC). Autrement dit, au lieu de rechercher des images par des meta-données, souvent du texte, le concept est de rechercher par le contenu des images. Extraire le contenu sémantique d'une image reste difficile et nécessite au préalable une extraction d'information de bas niveau. La majorité des systèmes RIBC sont fondés sur l'extraction d'information de bas niveau, où trois catégories d'information se sont démarquées. Les systèmes comme QBIC (Flickner et al., 1995), VisualSeek (Smith & Chang, 1997) et PhotoBook (Pentland et al., 1994) extraient l'information de couleur, texturale et géométrique pour constituer leurs primitives.
Cependant ces systèmes ne permettent pas des requêtes de recherche de haut niveau. Dans le cadre de la gestion d'énormes bases de données, le paradigme de la fouille d'information est pour la première fois formalisé par Keim & Kriegel (1996). La fouille d'information est le processus engagé pour explorer et découvrir des connaissances à partir de grandes quantités d'informations stockées dans des bases de données. Cette approche permet aussi de rechercher par le contenu tout en intégrant la connaissance du chercheur d'information. Dans cette approche, l'extraction d'information prend un caractère échelonnable, où l'information passe continûment d'un bas niveau vers un haut niveau. Cette information est tout d'abord représentée par les primitives (couleurs, textures, formes), ensuite par la sémantique (forêt, ville, chien, visage, ...) et enfin par la connaissance (phénomène atmosphérique, visages souriants, ...). La sémantique étant liée au chercheur d'information, elle est prise en compte lors du processus de recherche en faisant interagir l'utilisateur avec le moteur de recherche. Aussi, le processus est généralement itératif et l'utilisateur signifie au moteur de recherche la pertinence de la recherche à chaque itération. En quelque sorte, par son intervention, le chercheur d'information fait apprendre au moteur de recherche une mesure de similarité adaptée à sa connaissance. Ce paradigme de la fouille de données présente l'avantage de combiner la flexibilité et

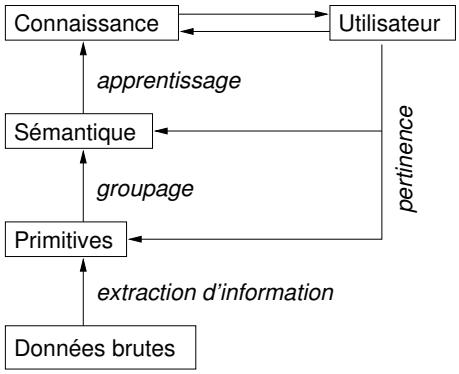


FIG. 2.2 – Organigramme du paradigme de la fouille d'information

la créativité de l'homme aux capacités énormes de stockage et de calcul des ordinateurs. Une esquisse de ce paradigme est présentée dans la Figure 2.2. Minka & Picard (1997) intègrent pour la première fois ce paradigme dans un système de fouille d'images nommé FourEyes s'appuyant sur les primitives de PhotoBook. Dans le cadre de la gestion d'archives d'images satellitaires, Seidel et al. (1997), Datcu et al. (1998), Schroder et al. (1998) présentent le système KIM intégrant le paradigme de la fouille de données. Par la suite, GeoIRIS (Shyu et al., 2007) se basant sur le même paradigme et tenant compte du contexte spatial a été développé. Dans cette vision du problème (Datcu et al., 2003), la recherche ou la fouille de données sont vues comme un canal de communication entre les données et un utilisateur. Ce concept, illustré dans la Figure 2.3, est la jonction de deux canaux de communication. Dans un premier temps, l'information est extraite objectivement des données. Cette information est encore appelée la représentation du signal. Ce premier canal de communication correspond à l'étape préliminaire d'extraction du moteur de recherche par le contenu présenté dans la Figure 2.1. Dans un second temps, l'information est extraite subjectivement sous l'action d'un utilisateur. Cette information est représentée par des modèles sémantiques et syntaxiques compréhensibles par le chercheur d'information. Des techniques d'apprentissage supervisé sont mises en place pour guider l'ordinateur vers les recherches les plus pertinentes pour l'utilisateur.

Dans le cadre de ce travail, nous nous intéressons uniquement à l'extraction d'information objective. Cette extraction d'information objective se fait en deux étapes majeures. La première étape consiste à extraire les primitives de chaque objet. La deuxième étape consiste à regrouper les primitives pour constituer des classes d'objets. Nous présenterons

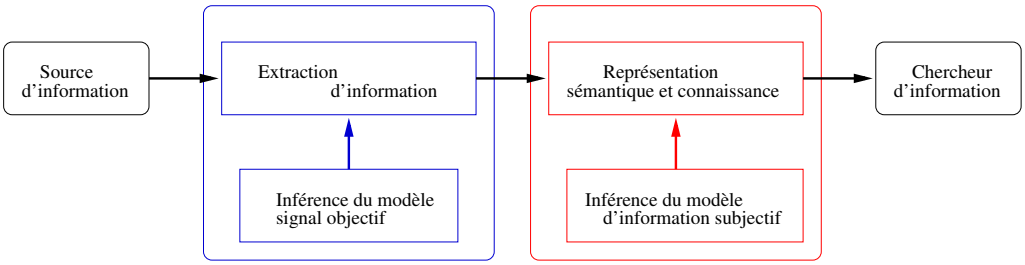


FIG. 2.3 – La fouille d'information vue comme un canal de communication. Les deux représentations, objective et subjective, communiquent pour former un canal de communication entre les données et un utilisateur.

dans un premier temps les différentes méthodes pour extraire les primitives des images. Ensuite nous présenterons comment les primitives sont extraites des vidéos et des séquences d'images satellitaires qui nous intéressent particulièrement dans ce travail. De plus, nous présenterons comment extraire l'information objective des données compressées. Dans un deuxième temps, nous présenterons les méthodes utilisées pour regrouper ces primitives et du même coup les objets numériques en classes homogènes. Enfin, nous terminerons sur un parallèle entre l'extraction d'information et la compression qui introduira notre nouveau concept pour l'indexation.

2.2 Extraction d'information des images

Pour décrire le contenu visuel d'une image, de nombreuses primitives ont été testées et/ou utilisées ces dernières années. Après dix années d'évolution, il résulte que ces primitives sont classées en trois catégories, chacune représentant un type d'information : la couleur, la texture et la forme.

2.2.1 L'information de couleur

Une des premières approches décrivant l'information de couleur a été d'utiliser les histogrammes de couleurs (Swain & Ballard, 1991) ou les moments de premiers ordres des distributions (Stricker & Orengo, 1995) tels que la moyenne et la variance. Ces deux descriptions présentent l'avantage d'être invariantes aux translations et rotations opérées sur les images. Cela permet ainsi de décrire deux images identiques à une transformation géométrique près par la même information de couleur. Néanmoins, deux images représentant des contenus différents peuvent partager le même histogramme. Pour s'affranchir de ce problème, Pass et al. (1996) distinguent les pixels appartenant à des régions uniformes ou non pour construire un histogramme de vecteurs de couleurs cohérentes et incohérentes. Cette approche intègre un peu d'information spatiale. Partant de cette dernière observation, Huang et al. (1997) utilisent les corrélogrammes de couleurs comme primitives, où le corrélogramme représente les corrélations de couleurs entre pairs de pixels séparés par plusieurs distances. Cette dernière extraction d'information améliore les résultats de recherche et elle est alors considérée comme meilleure. Enfin, une méthode est présentée par Mojsilovic et al. (2002) pour extraire les couleurs les plus importantes contenues dans une image fondée sur une quantification de l'espace des couleurs Teinte Saturation Valeur (TSV). Cette méthode a pour avantage de calculer des primitives moins complexes que des histogrammes tout en conservant la même information de couleur sans toutefois intégrer les relations spatiales.

2.2.2 L'information texturale

La texture s'est avérée une information importante pour la caractérisation du contenu d'une image. Fondamentalement la texture est définie comme la répétition d'un motif créant une image visuellement homogène. Plus précisément, la texture peut être vue comme un ensemble de pixels (niveaux de gris) spatialement agencés selon un certain nombre de relations spatiales, ainsi créant une région homogène. De ces définitions, les recherches sur la modélisation des textures se sont portées sur la caractérisation de ces relations spatiales.
Comme pour la description de l'information de couleur, Haralick et al. (1973) utilisent

des mesures statistiques pour discriminer les structures des images. Ils proposent quatorze primitives calculées à partir d'une matrice de co-occurrences qui correspondent à des statistiques de second ordre. Il faut attendre une vingtaine d'années pour voir émerger de nouvelles primitives. En outre, de nombreuses approches utilisent des transformées pour extraire l'information de texture. Jain & Farrokhnia (1991); Manjunath & Ma (1996) introduisent les filtres de Gabor à deux dimensions pour extraire l'information texturale en calculant les moyennes et les énergies des sorties de chaque filtre. Sur le même principe d'autre transformées sont ensuite étudiées comme les décompositions en ondelettes (Daubechies, 1990), en contourlet (Do & Vetterli, 2003), en paquets d'ondelettes (Laine & Fan, 1993), en arbre structurant d'ondelettes (Chang & C.-C.J., 1993), par filtres miroirs en quadrature (Randen & Husoy, 1995), où les moyennes et variances des réponses de sorties constituent les primitives. Ces espaces transformées sont utilisés pour la caractérisation des textures puisqu' ils décrivent les fréquences et les orientations des textures. Dans l'étude comparative effectuée par Randen & Husoy (1999), les primitives extraites par les filtres de Gabor et miroirs en quadrature donnent les meilleures caractérisations des textures. Ces modélisations sont ensuite améliorées par l'utilisation de variables de Markov cachées pour bénéficier des relations interbandes (Choi & Baraniuk, 2001; Do & Vetterli, 2002) et pour construire des primitives invariantes aux transformations géométriques. Ces invariances ont leur importance car elles permettent de caractériser des textures indépendamment du point de vue où elles sont observées. Ce dernier concept a fait apparaître une nouvelle famille de descripteurs de structure invariants par transformations affines et par échelle (Lazebnik et al., 2003).
En parallèle une autre famille de modèles de textures est apparue. Cross & Jain (1983a) introduisent les champs aléatoires de Gibbs-Markov pour modéliser les dépendances spatiales existant dans une texture. Ce type de modèle spécifie qu'un pixel dépend uniquement d'un voisinage donné, où les dépendances statistiques du voisinage sont calculées et constituent les primitives. Les champs de Gibbs-Markov sont en fait des modèles statistiques paramétrés dont les paramètres sont estimés. D'ailleurs, Geman & Geman (1984) montrent qu'il est possible de synthétiser des textures semblables à partir d'un jeu de paramètres et démontrent ainsi l'importance des champs aléatoires pour l'extraction l'information texturale. Par exemple, Chellappa & Kashyap (1983, 1985) présentent les processus simultanément auto-régressifs et les champs aléatoires de Gauss-Markov qui sont des cas particuliers de champs de Markov-Gibbs. La décomposition de Wold (Liu & Picard, 1996) fait aussi partie de cette famille et prend en compte le désordre existant dans la texture. Schroder et al. (1998) introduisent les champs autobinomiaux pour caractériser les textures dans les images satellitaires optiques. Les auteurs démontrent la supériorité de ce modèle par rapport aux autres champs puisqu'il nécessite moins de paramètres tout en décrivant correctement les textures.
Tandis que les champs de Gibbs-Markov modélisent les interactions spatiales localement, les transformées capturent des relations spatiales plus étendues. D'autre part, lors de l'extraction d'information, les champs aléatoires créent moins de primitives que les transformées, ainsi réduisant les complexités de calcul lors des recherches par le contenu. Enfin, Schröder & Dimai (1998) présentent la supériorité des champs aléatoires de Gibbs-Markov sur les transformées de Gabor pour caractériser les textures. Même si les transformées sont majoritairement utilisées pour extraire l'information texturale, il est préférable d'utiliser les champs de Gibbs-Markov produisant de meilleurs modèles.

2.2.3 L'information géométrique

L'extraction d'information géométrique a été le fer de lance de la recherche d'image par le contenu ces dernières années. Même si la caractérisation du contenu géométrique s'est avérée complexe plusieurs primitives géométriques ont montré leurs intérêts dans des systèmes de recherche.
Le premier système à intégrer la géométrie pour la recherche fût QBIC (Flickner et al., 1995). Des primitives caractérisant les formes sont utilisées telles que l'aire, la circularité, l'excentricité, les axes majeurs et les moments algébriques invariants. Néanmoins avant de pouvoir extraire les primitives de formes, il est nécessaire d'extraire ces formes par segmentation de l'image. Skarbek & Koschan (1994), Lucchese & Mitra (2001) donnent une vue générale des techniques de segmentation. Ils distinguent les méthodes de segmentation pixellaire et fondée sur les régions. Les méthodes de segmentation pixellaires utilisent essentiellement l'information de couleur pour constituer des régions de couleurs homogènes. D'autre part les méthodes fondées sur les régions restent plus efficaces car elles s'appuient sur l'information de texture et/ou la détection de contours. Deux sous-types de méthodes se distinguent. L'un concerne les méthodes par croissance de régions à partir de graines, tandis que l'autre concerne les méthodes par fusion et partage de régions. Les techniques de segmentation qui se sont le plus démarquées ces derniéres années sont la segmentation par contours actifs (Rochery et al., 2003), le Mean Shift (Comaniciu & Meer, 2002), les coupes de graphes (Shi & Malik, 2000) et les modèles à variables latentes (Carson et al., 2002). Une fois la segmentation obtenue, les primitives de forme sont extraites. Veltkamp & Hagedoorn (1999) présentent un état de l'art sur l'analyse et la mise en correspondance des formes. Néanmoins, cette approche fondée sur la segmentation est très discutable du fait que les relations spatiales inter-régions ne sont pas considérées. Pour pallier ce problème, deux mesures de similarité sont communément employées. La mesure Integrated Region Matching (Wang et al., 2001) permet de mettre en correspondance toutes les régions de deux images et la mesure Earth Mover's Distance (Rubner et al., 1998) permet de mesurer le coût nécessaire pour transformer un ensemble de régions en un second. D'autre part, des modélisation par des graphes de régions (Matas et al., 1995) permettent de modéliser l'agencement des régions entreelles. Par exemple, Li & Bretshneider (2007) utilisent des réseaux bayésiens définis sur ces graphes pour extraire l'information d'images satellitaires. Même si ces représentations paraissent recueillir de l'information de plus haut niveau, elles créent des primitives plus complexes que les primitives texturales et de couleur, ainsi réduisant la vitesse de recherche.

2.3 Extraction d'information des séquences d'images

Comme pour les images, les séquences d'images peuvent être caractérisées par leur contenu. Seulement, la dimension temporelle est à prendre en compte. En conséquence, en plus de l'informations de couleur, texturale et géométrique, l'information de mouvement est considérée pour la caractérisation et l'indexation des séquences d'images. Le type de séquences d'images qui a suscité le plus d'intérêt ces dernières années, est la vidéo. En effet avec l'augmentation des capacités des réseaux, le partage et la distribution de vidéos sont en plein essor. Nous présenterons dans la suite, l'extraction d'information des vidéos dans un premier temps. Dans un second, nous présenterons les méthodes d'extraction de primitives sur les Séries Temporelles d'Images Satellitaires qui

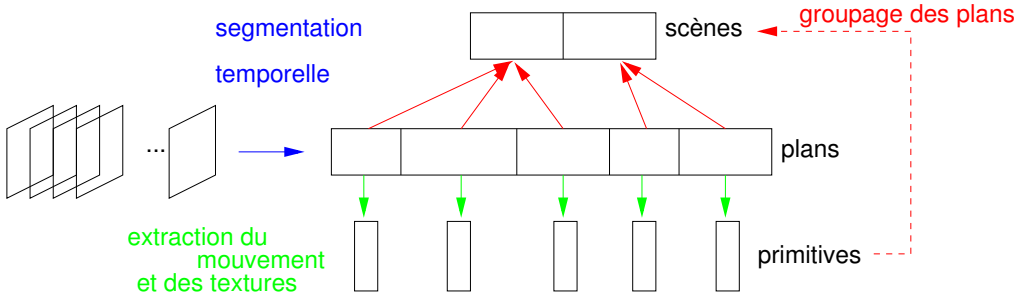


FIG. 2.4 – L'extraction d'information dans les vidéos exploite leurs constructions. La vidéo est alors représentée par des scènes, puis des plans qui sont eux-mêmes représentés par une information de mouvement ou une information liée à une image représentative.

nous intéressent plus particulièrement.

2.3.1 Information de mouvement dans les vidéos

La norme MPEG-7 s'est intéressée très tôt à la normalisation d'une description des vidéos qui inclut le son et la partie visuelle. Elle permet en particulier la création d'une description du contenu visuel, comprenant selon la volonté de l'utilisateur, un niveau de détail variable. Les vidéos ou les films contiennent potentiellement énormément d'information par rapport aux images. A titre d'exemple une vidéo de 5 min ne contient pas moins de 7500 images à raison de 25 images par secondes. Pour ces raisons, une vidéo est vue généralement comme un assemblage hiérarchique de scènes et de plans. Un plan correspond à un événement qui se produit dans un certain fond, tandis que la scène est une juxtaposition de plans dont la logique permet d'exprimer une idée plus élaborée. En conséquence, les vidéos sont segmentées temporellement en plans pour conduire l'analyse et l'extraction d'information. Au même titre que les techniques de segmentation spatiale utilisées pour la caractérisation des objets d'une image, la segmentation temporelle permet de partitioner une vidéo en une suite d'objets événementiels. Ensuite, des méthodes d'extraction plus spécifiques sont utilisées sur ces différents plans pour décrire les événements. La Figure 2.4 présente cette structuration hiérarchique pour l'extraction d'information. Premièrement, nous aborderons les méthodes de segmentation temporelle, puis nous présenterons la caractérisation des plans.
Plusieurs types de transitions entre plans (fondu, dissolution) peuvent être utilisées lors du montage d'une vidéo, ce qui rend plus complexe la segmentation par rapport à un changement abrupte de plan. Koprinska & Carrato (2001) présentent une vue générale des méthodes de segmentation temporelle qui vont des modèles de Markov caché aux techniques utilisant directement les vecteurs de mouvements et les coefficients des transformées des flux MPEG. Ensuite ces différents plans peuvent être regroupés dans des scènes, en considérant des similarités entre plans ou en modélisant des suites logiques d'événements (Hanjalic et al., 1999). Cette extraction d'information est de plus haut niveau et la plupart des méthodes d'extraction d'information se cantonnent à la segmentation temporelle de plans.
Plusieurs approches ont été expérimentées pour extraire des primitives pertinentes d'un plan. Une première approche consiste à sélectionner l'image ou les images les plus représentatives et d'y appliquer l'extraction d'information (Yueting et al., 1998). Néanmoins ces méthodes restent inefficaces car elles n'utilisent pas l'information temporelle. Par

conséquent, d'autres méthodes se sont attachées à extraire l'information de mouvement. Par exemple, les vecteurs de mouvements obtenus lors de la compression sont les premiers paramètres qu'il est possible d'extraire. Ces vecteurs sont calculés par une estimation de mouvements se basant sur la conservation du flux optique dans le temps (Barron et al., 1994). A partir de ces vecteurs de mouvements, les histogrammes des directions des mouvements sont utilisés comme primitives par Kobla et al. (1997) pour différencier les plans avec beaucoup ou peu d'actions. Nous trouvons aussi l'estimation des mouvements de caméra par des transformations affines. En effet, ces paramètres servent à indiquer l'existence de zooms, de rotation et de translations. Enfin, la segmentation spatio-temporelle et le suivi des objets sont utilisés pour la caractérisation des plans. Cette représentation des plans est celle la plus en adéquation avec la description qu'un homme ferait. Toutefois l'indexation des objets et de leur mouvement demeure rare. VideoQ (Chang et al., 1998) fait partie de ces systèmes et créer l'index des trajectoires des objets. D'autres méthodes plus perfectionnées ont vues le jour depuis pour la segmentation spatio-temporelle, où l'accent est mis sur la détection des objets et de leur mouvement à partir d'inférence de modèles statistiques tels que les champs de Markov à variables latentes (Wang & Loe, 2005).
Finalement, en marge des méthodes de segmentation spatio-temporelle, des études se sont penchées sur la modélisation de texture temporelle où le mouvement n'existe pas. Par exemple, Rahman & Murshed (2004) généralisent la méthode de Haralick et al. (1973) en utilisant les matrices de co-occurrence en trois dimensions pour extraire l'information liée à l'évolution d'une texture. Cependant, cette extraction d'information est adaptée à des vidéos particulières et peu répandues comme la pluie.

2.3.2 Méthodes d'extraction dédiées aux séquences d'images satellitaires

Les intérêts des séquences d'images satellitaires sont bien différents des vidéos et ont amené d'autres types d'analyse et de moyens d'extraction. Tout d'abord, ces séquences ne contiennent pas beaucoup d'images à cause de la faible couverture spatio-temporelle des satellites d'acquisition. D'autre part, les images sont alignées durant la construction des séquences et par conséquent le mouvement y est inexistant. De ce fait, beaucoup d'études se sont focalisées sur la détection des changements entre deux images consécutives, puisque les changements sont importants dans des applications de suivi telles que la surveillance de l'environnement et des forêts, le contrôle de l'agriculture et l'extension des zones urbaines. La détection de changements consiste à construire une carte spatiale présentant les pixels ayant évolués ou non. Beaucoup de méthodes de détection de changements ont été appliquées aux images Radar à Synthèse d'Ouverture (RSO). De plus, les études ont été poussées sur la détection automatique, pour tirer parti au mieux de l'énorme quantité d'images.
Nous trouvons les méthodes par analyse en composantes principales, où les changements apparaissent sur les composantes de faible énergie. Bovolo & Bruzzone (2005) présentent une méthode fondée sur la différence des log-images, où cette image différence est analysée par une décomposition en ondelettes. Ensuite chaque pixel est classifié en fonction de plusieurs échelles et en comparant ces statistiques locales aux statistiques globales. Cette méthode permet de prendre en considération l'échelle des objets de manière automatique. Bazi et al. (2005) proposent une approche bayésienne, en modélisant l'appartenance d'un pixel à une des deux classes par des mélanges de lois statistiques. Ces méthodes sont spécifiques aux images RSO, puisqu'elles intègrent le débruitage du *spe-*

ckle tout en limitant la perte de précision géométrique dans l'image. D'autres approches consistent à comparer les segmentations des images opérées aux deux instants. Giros (2006) propose une méthode permettant d'inférer une segmentation commune à deux ou plusieurs segmentations. En comparant la segmentation commune aux autres segmentations, les changements peuvent être détectés. Seulement, ces méthodes sont limitées par la dimension temporelle et ne permettent pas une extraction d'information pour la gestion d'une base de séquences multitemporelles d'images. En outre, ces méthodes détectent les changements abrupts et non les changements progressifs qui s'opèrent sur plusieurs images.
Beaucoup de méthodes qui essaient de modéliser les changements progressifs dans les séquences ne prennent pas en compte la dimension spatiale. Par exemple, Aurdal et al. (2005) utilisent les chaînes de Markov cachées pour modéliser une certaine évolution contrainte par la phénoménologie. Sur le même principe, Kawamura et al. (2004) décrivent comment extraire des règles de dépendances temporelles à partir d'une sémantique spatiale. Encore, Jing et al. (2005) utilisent l'évolution de l'Indice de Végétation Différentiel Normalisé (IVDN) pour le suivi des zones agricoles. Cet indice est surtout utilisé dans les images multi-spectrales de faible résolution. Finalement, Heas & Datcu (2005) décrivent une méthode plus complexe pour l'extraction d'information dans les STIS à haute résolution où les informations spatiale et temporelle sont utilisées conjointement. Ils proposent une méthode bayésienne où plusieurs niveaux d'information sont extraits. En particulier ils se placent dans le concept de la fouille d'information et proposent une modélisation hiérarchique qui va des données vers l'utilisateur. Nous présentons uniquement l'extraction d'information non supervisée qui constitue la première brique du système. Cette extraction est l'inférence de graphes qui modélisent des trajectoires de clusters (cf. §2.5.2) dynamiques qui codent les structures spatio-temporelles. Cette modélisation exploite la constatation que l'analyse doit être temporellement et spatialement localisée. Ainsi, les structures spatio-temporelles sont décrites par des modèles paramétriques et des mélanges de lois. Ensuite, pour prendre en compte l'évolution dans les STIS, les structures sont reliées temporellement à l'aide de graphes.
Ce même genre d'idée a été repris dans ce travail (Gueguen et al., 2006). Dans cet article, deux approches sont comparées. La première approche essaie de regrouper des trajectoires de régions, où la similarité est fondée sur l'information de texture. L'autre approche modélise l'évolution de régions spatiales prédéfinies en se basant sur une segmentation commune à toutes les images (Giros, 2006).

2.4 Extraction d'information des données compressées

En parallèle des études sur la fouille de données, le domaine de la compression d'images et des vidéos s'est largement développé pour répondre aux besoins d'échange sur des canaux à capacité limitée et aux besoins de stockage. Avec l'accroissement des bases d'images et de vidéos, cet intérêt s'est avéré encore plus important. En effet de nos jours, il est rare de trouver une image/vidéo sous forme décompressée. Ainsi, de nombreuses études se sont penchées sur l'extraction d'information dans les données compressées. Le problème se complexifie, car pour gagner en temps de calculs, il faut extraire l'information sans décompresser totalement l'objet codé. Nous présentons, l'extraction d'information dans le texte compressé qui fût au point de départ de la problématique. Ensuite, nous discutons des méthodes d'extraction d'information des images et vidéos compressées.

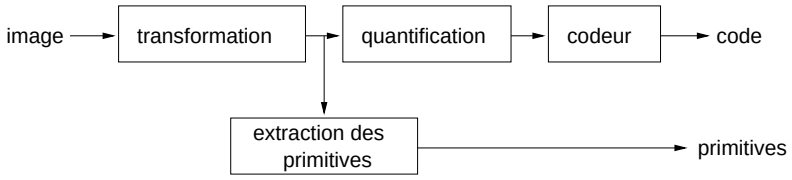


FIG. 2.5 – Cet organigramme présente la procédure de compression avec pertes d'images et l'extraction d'information. Nous observons que le code et les primitives sont deux objets dissociés. De plus si les primitives ne sont pas calculées, il faut décoder et obtenir la transformée pour extraire l'information.

2.4.1 Extraction d'information dans le texte compressé

Les méthodes les plus avancées pour la recherche de motif dans un flux compressé concernent le texte. Ces méthodes sont optimales en temps de recherche et en espace de stockage. Farach & Thorup (1998) proposent un algorithme pour retrouver un motif dans un flux compressé par le codeur universel Lempel-Ziv-Welch (Welch, 1984). Cet algorithme ne nécessite pas le décodage de toute la séquence lors de la recherche d'un motif. Pajarola & Widmayer (1996) étendent le concept au cas de signaux bidimensionnels où ces signaux sont encodés ligne par ligne. L'algorithme permet aussi de retrouver des motifs bidimensionnels dans le flux sans le décoder. Néanmoins, ce type de compression est loin d'être optimal car elle ne prend pas en compte les relations spatiales entre lignes. Enfin Grossi & Vitter (2005) présentent un algorithme de compression de texte fondé sur la transformation de Burrows-Wheeler qui donne une représentation permettant des recherches encore plus rapides. De plus, ils présentent le compromis qui existe entre l'efficacité de compression et la complexité algorithmique des recherches. Ils remarquent que si l'efficacité de compression diminue, alors la complexité de recherche augmente. Ce compromis est souvent peu discuté théoriquement dans la création de système de recherche, et le problème se règle empiriquement. Enfin, le désavantage de toutes ces méthodes est qu'elles sont dédiées à la recherche exacte de motifs dans des chaînes encodées. Par conséquent, ces méthodes ne sont pas adaptées pour la recherche inexacte dans des bases de signaux multidimensionnels qui sont compressés différemment des signaux unidimensionels.

2.4.2 Extraction d'information des images compressées

Mandal et al. (1999) passent en revue l'extraction de primitives des images dans le domaine compressé. Ils présentent plus particulièrement les méthodes d'extraction à partir des standards et des techniques de compression communément utilisées. On retrouve le codage par Transformée en Cosinus Discret (TCD) dans JPEG et par transformée en ondelettes dans JPEG2000, où l'extraction de primitives est fortement semblable aux méthodes du §2.2.2. Cette extraction est présentée dans l'organigramme de la Figure 2.5. Nous remarquons que l'image doit être décodée pour extraire les primitives.
Zhang et al. (1995) proposent une extraction d'information des images fondée sur le codage fractal introduit par Jacquin (1993). Ce codage exploite les autosimilarités contenues dans l'image à différentes résolutions. Les Systèmes de Fonctions Itérées (SFI), sur lesquelles s'appuie le codage fractal, constituent les primitives et représentent le code de l'image. Il est montré que ces primitives sont aussi performantes que les primitives extraites des transformées en ondelettes. Les SFI sont caractérisés par plusieurs paramètres

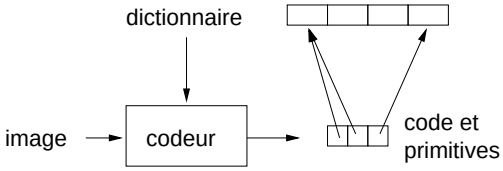


FIG. 2.6 – L'organigramme présente l'extraction d'information et le codage conjoint fondés sur un dictionnaire. Cette compression est avec pertes. Nous notons que le code constitue l'information extraite. Le code est composé des indices des mots de code du dictionnaire.

dont les paramètres d'échelle et de luminance. Schouten & Zeeuw (2000) montrent que l'utilisation d'histogrammes des paramètres d'échelles donne d'aussi bons résultats que les SFI. Ces histogrammes donnent la répartition des échelles existant dans l'image. Nous noterons que l'échelle est un paramètre important pour la caractérisation des textures (Luo et al., 2006) et est fortement lié à la rugosité. Enfin, Pi et al. (2005) améliorent cette extraction en utilisant des histogrammes des paramètres d'échelle et des paramètres de luminance représentant l'information de couleur. Ils prouvent de plus que ces deux types de paramètres sont indépendants. Il est à noter que les techniques de codage précédentes caractérisent essentiellement l'information texturale.
Un autre technique de compression couramment utilisée est la quantification vectorielle (Gray & Neuhoff, 1998). Zhu et al. (2000) compressent les images à partir de blocs-clé. A partir d'une base d'apprentissage, un dictionnaire de blocs-clé est créé où chaque bloc est associé à un indice. Quand une image est compressée, celle-ci est partitionnée en blocs, puis ceux-ci sont codés par les indices des blocs-clé les plus proches. Ainsi la collection d'indices constitue à la fois le codage et les primitives de l'image. La Figure 2.6 schématise l'extraction d'information et le codage par dictionnaire des images. Chaque bloc-clé peut être vu comme un mot et l'image comme une collection de mots. Alors, les techniques de fouille d'information dans le texte peuvent s'appliquer directement à cette représentation. Cette même méthodologie a été employée sur les coefficients de transformée en ondelettes (Idris & Panchanathan, 1995) et sur les coefficients TCD (Podilchuk & Zhang, 1996) pour exploiter au mieux l'information texturale. Les techniques de compression fondées sur la quantification vectorielle ont un avantage certain en fouille d'image sur les autres techniques puisque la quantification vectorielle est une méthode d'indexation naturelle (cf. §2.6.3), composante importante d'un système de recherche. Néanmoins, la quantification vectorielle n'est pas aussi efficace en compression que les techniques fondées sur les transformées. En outre, la dimension et la création du dictionnaire sont des facteurs cruciaux pour un codage efficace. Comme nous l'avons remarqué toutes ces méthodes extraient l'information texturale. Aussi, Swanson et al. (1996) essaient d'intégrer l'information géométrique dans le codage de l'image. L'image est tout d'abord segmentée, puis chaque région est recouverte par un ensemble de blocs. Ensuite, chaque région est codée par blocs-clé de coefficients TCD, où la position spatiale de ceux-ci est stockée. Lors du codage, les auteurs introduisent un critère faisant le compromis entre la complexité de recherche et l'efficacité de compression. Ainsi, lors d'une requête, une petite partie du flux est décompressée en moyenne et l'information géométrique est prise en compte.
Toutes les méthodes d'extraction précédentes sont fondées sur des compressions avec pertes des images/vidéos. Jiang et al. (2003) présentent une extraction de primitives dans les images compressées par JPEG-LS (Weinberger et al., 2000) qui est un codeur sans

perte. Ils utilisent les histogrammes des états des contextes comme primitives. Ils montrent expérimentalement que ces primitives caractérisent mieux les textures que les autres primitives extraites des espaces transformées dans le cas où la compression est avec pertes. Il apparaît qu'une compression sans perte est préférable pour l'extraction de primitives. En fait, la force de JPEG-LS est liée au modèle statistique paramétré qui est inféré lors du codage. Pajarola & Widmayer (2000) présentent aussi une technique d'extraction similaire pour la compression et l'extraction d'information dans les images satellitaires, où les images sont compressées sans perte. Ces techniques de codage sont regroupées sous le nom de Modulation Différentielle d'Impulsions Codées (MDIC) où le signal est prédit et l'erreur résiduelle est codée (cf. §2.6.2).
Enfin, nous trouvons des méthodes d'extraction encore marginales mais présentant des bons résultats. Tabesh et al. (2005) prend comme primitives les longueurs de code de chaque bande de la transformée en ondelettes de JPEG2000. Cette information est facilement accessible dans les entêtes du flux compressé. Ces primitives sont comparées aux énergies des sous bandes et elles donnent de meilleurs résultats pour caractériser les textures. Ces longueurs mesurées en bits sont en fait fortement corrélées aux logarithmes des énergies des sous bandes.
Nous remarquons que les techniques de compression permettent d'extraire principalement l'information texturale. D'autre part, il semble que les compresseurs sans perte permettent d'extraire mieux cette information. Enfin, nous observons que le compromis entre capacité de compression et complexité de recherche est évincé dans la plus part des cas. Toutefois, nous avons l'intuition qu'il est une des clés pour la création de systèmes de recherche par le contenu efficaces.

2.4.3 Extraction d'information des vidéos compressées

Les standards internationaux tels que MPEG-1 et MPEG-2 ont connu un énorme succès dans l'industrie de l'information. D'autres standards plus performants tels que MPEG-4 et H.26x partagent les fondements des premiers standards. Par conséquent, les techniques d'extraction d'information varient peu d'un standard à l'autre. Tout d'abord, la compensation de mouvement est une composante commune à tous les standards. Dans le domaine compressé, l'utilisation des vecteurs de mouvements est privilégiée aux flux optiques, puisque cette information est directement accessible. Wang et al. (2003) présente une vue générale de l'extraction des primitives de mouvements (cf. §2.3.1) : mouvements de caméra, trajectoires d'objets, quantité de mouvements. En particulier, Kobla et al. (1999) présentent une extraction qui utilise les vecteurs de mouvements avec le nombre de bits nécessaire au codage de chaque image pour la détection de ralenti. L'utilisation de la longueur de codage montre encore son efficacité par rapport à la simplicité d'accès de cette caractéristique.

2.5 Mesures de similarité et création de l'index

Dans les paragraphes précédents, nous avons présenté les méthodes qui permettent de décrire le contenu des images ou des séquences par des primitives. Ce processus génère souvent une grande quantité de données qui ne peut être gérée en pratique, particulièrement si plusieurs types de primitives (couleur, texture, forme, mouvement) sont agrégés. Pour pallier ce problème, certains systèmes intègrent des méthodes de réduction des primitives. De plus, si lors de la recherche, chaque objet de la base est comparé à la

requête alors les temps de recherche peuvent devenir prohibitifs dans le cas d'énormes bases de données. Pour réduire cette complexité de recherche et obtenir un système de recherche s'adaptant à une base grandissante, des méthodes d'indexation ou de groupage ont été élaborées. Comme, il est signalé au §2.1, la définition ou le calcul d'une mesure de similarité a une importance majeure pour la création de l'index. En effet, lors de l'indexation les objets sont groupés par similarité. Ainsi, lors d'une requête et grâce à l'index, seuls quelques groupes et leurs membres sont comparés à la requête ce qui réduit considérablement les temps de recherche.

2.5.1 Réduction des primitives

Les méthodes de réduction de primitives sont particulièrement adaptées aux primitives représentées sous forme de vecteurs de dimension finie m. Le but des méthodes de réduction de dimensionnalité consiste à transformer l'espace des primitives en un espace de dimension plus petite $k < m$, tout en conservant au maximum l'information. Une méthode commune est d'appliquer la transformée de Karhunen-Loéve (Duda et al., 2001) et d'éliminer les composantes avec une faible variabilité. Ce type de transformation permet de décorreler les composantes et de les rendre indépendantes sous certaine conditions de gaussianité. Ainsi, les composantes avec une faible variabilité sont d'une part peu discriminantes et d'autre part contiennent peu d'information. Par conséquent, leur élimination diminue très peu le pouvoir de discrimination et du même coup l'information. Sur le même concept, d'autres méthodes d'analyse sont utilisées telles que l'analyse par poursuite de projection ou par composantes indépendantes (Lee et al., 2000) qui relâchent les hypothèses fortes de gaussianité.
D'autres méthodes consistent à supprimer certaines composantes redondantes des primitives. Ces techniques sont utilisées pour des primitives constituées de nombreuses composantes. Le principe général consiste à regrouper les composantes en plusieurs ensembles et à conserver la composante la plus représentative du groupe. Ainsi, une réduction de la dimension des primitives peut être obtenue. Campedel et al. (2004) présentent et comparent plusieurs méthodes de sélection de primitives, où la méthode fondée sur *Support Vector Clustering* (Mitra et al., 2002) donne les meilleurs résultats pour la caractérisation d'images satellitaires.
Enfin, la dernière approche consiste à limiter le nombre de primitives lors de l'extraction d'information par une pénalisation de la dimensionnalité (Rissanen, 1983). Cette approche s'applique particulièrement aux modèles stochastiques paramétrés et s'avère particulièrement efficace et concurrente des autres méthodes de réduction. D'autre part Globerson & Tishby (2003) présentent une approche qui consiste aussi à limiter les primitives, c'est-à-dire les paramètres, aux statistiques suffisantes qui permettent la description des observations. Ce concept est emprunté aux idées de Rissanen (1983).

2.5.2 Indexation non-supervisée et supervisée

L'indexation multidimensionnelle débuta originellement avec les quadtree développés pour les bases de données traditionnelles. QBIC fut le premier système à utiliser l'indexation pour diminuer les temps de recherche, où les paramètres de texture et de couleur sont indexés par un R-tree (Beckmann et al., 1990) qui partitionne hiérarchiquement un espace multidimensionnel. Ainsi les objets sont représentés par l'indice du groupe auquel ils appartiennent. Alors, lors d'une recherche, les similarités des groupes avec la

requête sont évaluées. Comme le nombre de groupes est moins important que le nombre d'objets, la recherche devient moins complexe. D'autre part l'utilisation d'arbres, tels que R-tree, permet de diminuer encore plus la complexité de recherche en diminuant le nombre de comparaisons hiérarchiquement. Nous présenterons dans la suite uniquement les méthodes d'indexation liées au groupage des objets par similarité, laissant l'aspect hiérarchique comme extension possible. Ces techniques d'indexation sont divisées en deux familles : supervisées et non-supervisées.

Les méthodes d'indexation supervisées, autrement appelées classification, exploite une base d'apprentissage pour construire un partitionnement de l'espace des primitives. La base d'apprentissage est composée d'objets qui sont déjà regroupés ou classifiés manuellement. Alors, un nouvel objet entré dans la base sera classifié ou indexé en fonction de l'ensemble de la partition auquel ses primitives appartiennent. Parmi les méthodes de classification majeures, il y a la classification par le plus proche voisin (Cover & Hart, 1967) qui est rendue plus robuste en considérant plusieurs voisins (Devroye, 1978). Une autre grande famille de classificateurs a été introduite par Fisher (1936) avec l'analyse discriminante linéaire qui a ensuite débouché sur les réseaux de neurones (McCulloch & Pitts, 1943). Akaike (1954) introduit les méthodes de classification à noyaux. Enfin, Vapnik (1982) tire parti de la théorie de Bayes pour inférer des classificateurs paramétrés qui ont ensuite amené la création des Machines à Vecteurs Supports (MVS) (Boser et al., 1992; Cortes & Vapnik, 1995) utilisant les noyaux et l'analyse discriminante avec des marges maximales. Cette dernière technique s'est avérée la plus performante en termes de classification.

Partitionner l'espace des primitives peut être vu comme l'inférence d'une mesure de similarité entre objets qui n'est pas donnée a priori. D'ailleurs cet aspect fait défaut aux méthodes des k-plus proches voisins où la distance est fixée. Fukunaga & Flick (1984) proposent alors d'utiliser des distances pondérées qui améliorent considérablement les résultats de classification. Par conséquent, le choix de la pondération correspond à la construction d'une mesure de similarité. En conclusion, les méthodes de classification essayent de créer une mesure de similarité pour grouper les objets en se référent à une base d'apprentissage. Néanmoins, ces méthodes ne sont pas adaptées à l'indexation de bases de données dans le cadre du paradigme de la fouille présenté au §2.1. Puisque la base d'apprentissage est construite par une personne ou est en lien direct avec une application, le classificateur qui en résulte est subjectif. Or le paradigme insiste sur le fait que l'extraction d'information et l'indexation doivent être objectives et uniquement liées aux signaux. D'ailleurs les méthodes précédentes peuvent être utilisées au sein du même paradigme pour la fouille d'information dans l'extraction subjective liée à une application ou un utilisateur (Heas, 2005; Costache et al., 2006).

Les méthodes d'indexation non-supervisée, encore appelées clustering, sont quant à elles totalement adaptées à l'extraction d'information objective. Contrairement aux méthodes supervisées, elles sont fondées sur la définition d'une mesure de similarité ou une distance entre objets. Cet aspect est discuté dans la suite. Concernant les méthodes de groupage, nous trouvons par exemple l'algorithme des k-moyennes discuté par MacQueen (1965) pour le regroupage de données multivariées. Lors de ce processus, chaque point de l'espace des primitives est représenté par le centre du groupe ou du cluster auquel il appartient. Alors, cette nouvelle représentation des objets constitue l'index. Cette méthode des k-moyennes a ensuite été étendue en k-moyennes floues où un objet appartient à un groupe avec une certaine probabilité (Bezdek, 1981). Ces méthodes font appel à des optimisations du type Espérance Maximisation. Durant, ces processus d'optimisations des

nouveaux points de l'espace des primitives sont créés par des combinaisons linéaires. Et dans le cas d'espaces plus complexes que les espaces vectoriels, il est souvent impossible de créer de nouveaux objets. Pour pallier ce problème, Kaufman & Rousseeuw (1990) décrivent l'algorithme des k-médoids qui à partir d'un matrice de distance est capable de regrouper les primitives autour d'objets déjà existants. Quand l'algorithme de k-moyennes calcule les moyennes des groupes, l'algorithme des k-médoids cherchent l'objet médian vis-à-vis de la similarité. Il existe aussi le regroupage hiérarchique (Johnson, 1967) qui permet de ne pas recalculer de nouveaux objets. Cet algorithme regroupe les objets deux à deux en remplaçant chaque objet par le groupe auquel il appartient. Ces algorithmes utilisent aussi uniquement la matrice de similarité entre objets pour obtenir le dendrogramme (Figure 2.7). Ensuite, pour obtenir l'index, il suffit de couper l'arbre à l'endroit désiré.

Ces algorithmes souffrent tous d'un même défaut qui est la détermination du nombre de groupes. En effet dans les algorithmes du type k-moyennes, le nombre de groupes est déterminé par k. Dans le groupage hiérarchique le nombre de clusters est aussi déterminé par la coupure de l'arbre. Ce paramètre a toutefois une grande importance puisqu'il détermine les performances et les complexités de recherche. Nous rappelons que les techniques d'indexation sont introduites pour diminuer la complexité de recherche. Cependant l'indexation entraîne une perte d'information qui diminue les performances de discrimination. Il faut donc obtenir des méthodes qui sélectionnent automatiquement le nombre de groupes pour conserver le maximum d'information tout en réduisant la complexité de recherche. Cheeseman et al. (1988) présentent un tel algorithme de groupage, AutoClass, dans le cadre Bayésien où le nombre de clusters est calculé automatiquement. Le défaut de cet algorithme est qu'il est coûteux en complexité de calculs. Sugar & James (1998) proposent un critère heuristique moins complexe lié à l'analyse débit-distorsion pour déterminer le nombre optimal de clusters.
Ces dernières méthodes de clustering se sont avérées les plus efficaces pour la construction de moteur de recherche par le contenu dans le domaine des images. Elles permettent de calculer des index qui accélèrent les temps de recherche tout en conservant l'information nécessaire à discriminer les objets de la base. D'autre part, ces méthodes rendent la description de la base de données robuste à son augmentation puisqu'un nouvel objet sera associé à un index de groupe déjà existant. Néanmoins, cette robustesse à l'échelle de la taille de la base est relative et limitée. Charikar et al. (1997) se sont attachés à développer des algorithmes de regroupage incrémentaux, où les groupes sont recalculés efficacement à chaque nouvelle entrée dans la base de données.

2.5.3 Mesures de similarité

La mesure de similarité est cruciale dans la conception du système de recherche. Comme nous l'avons vu précédemment, cette mesure est utilisée dans l'étape de groupage non-supervisée. Les mesures de similarité quantifient sur l'espace des réels les ressemblances entre objets. Leurs mesures duales associées sont les mesures de différence, telles que les distances. Quand deux objets sont similaires, la mesure de similarité est grande tandis que la mesure de différence est petite.
Les mesures les plus utilisées sont les distances puisqu'elles rassemblent les propriétés de symétrie, de séparation et d'inégalité triangulaire. Même si la dernière propriété d'inégalité

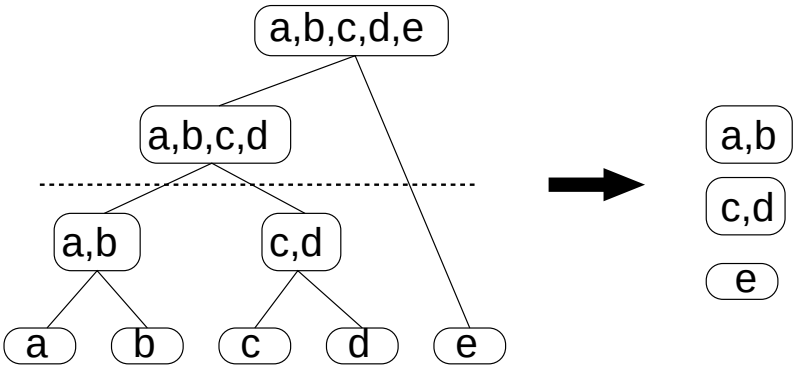


FIG. 2.7 – Le groupage hiérarchique consiste à regrouper les éléments deux par deux. Cette procédure est ascendante car elle part du niveau le plus bas de l'arbre ou dendrogramme vers la racine. Dans cet exemple, l'arbre est coupé tel qu'il résulte trois groupes pour construire l'index. Enfin, cet arbre peut être vu comme une codage préfix des éléments de l'ensemble. Par la construction fondée sur des groupages deux à deux aucun mot de code n'est le préfix d'un autre.

triangulaire est moins importante, les deux autres paraissent évidentes.

$$\begin{array}{lll} \forall x, y & d(x,y) = d(y,x) & \textit{(symétrie)} \\ \forall x, y & d(x,y) = 0 \Leftrightarrow x = y & \textit{(séparation)} \\ \forall x, y, z & d(x,y) \leq d(x,z) + d(z,y) & \textit{(inégalité triangulaire)} \end{array}$$

Si la symétrie est vérifiée, alors la distance ne dépend pas de l'ordre dans lequel sont comparés les objets. Si la propriété de séparation est vérifiée alors tout objet peut être discriminé, c'est-à-dire que tous les objets non égaux partagent une distance strictement positive. Dans le cas des moteurs de recherche, cette condition peut être élargie par :

$$\forall x, y \qquad d(x,x) \leq d(x,y)$$

Cette propriété peut être vue comme le fait que deux objets différents ne peuvent être plus similaires que l'objet est similaire à lui-même.
La mesure la plus répandue sur les espaces de primitives inclus dans un espace vectoriel de dimension fini tel que $\mathbb{R}^k$ est la distance euclidienne. En effet, c'est une distance naturelle associée à un espace vectoriel. Dans le cadre de l'indexation, nous avons noté que cette distance peut être étendue par une pondération des axes de l'espace. Cependant les espaces vectoriels ne constituent qu'une petite part des primitives que l'on peut extraire des images/vidéos. Alors, dans chaque cas, il faut trouver une mesure associée à l'information extraite. Dans le cas où les primitives sont des ensembles d'un espace vectoriel, la distance de Hausdorff est particulièrement adaptée. Rucklidge (1996) montre que cette distance donne de bons résultats en reconnaissance de formes où plusieurs paramètres locaux sont extraits de l'image. Comme le nombre de paramètres locaux varie d'une image à l'autre, il est plus judicieux de voir ces primitives comme des ensembles. Il existe aussi d'autres distances pour la comparaison de chaînes (Wagner & Fisher, 1974) et de graphes (Bunke & Allerman, 1983).
Finalement, nous trouvons des distances liées aux modèles générateurs englobés dans la modélisation de Bayes. Kullback & Leibler (1951) introduisent une mesure de divergence entre distributions statistiques. Dans le cas où il y a deux objets x_1 et x_2 modélisés par leur paramètres estimés θ_1 et θ_2 liés au modèle $p_X(X \mid \theta)$, la divergence entre ces deux

objets est définie au travers de leurs paramètres par :

$$d(\theta_1, \theta_2) = d_{KL}(p_X(X \mid \theta_1) \mid p_X(X \mid \theta_2)) = \sum_x p_X(X = x \mid \theta_1) \log \frac{p_X(X = x \mid \theta_1)}{p_X(X = x \mid \theta_2)}$$

Ce n'est pas une distance puisqu'elle ne vérifie pas la symétrie. Par contre, on peut définir la distance de Kullback-Leibler entre deux distributions p_X et q_X par $D_{KL}(p_X, q_X) = d_{KL}(p_X \mid q_X) + d_{KL}(q_X \mid p_X)$. C'est une distance dans le cas où le modèle est identifiable, c'est-à-dire qu'une seule et même distribution ne peut être générée par deux paramètres différents. Cette distance est aussi adaptée aux représentations par histogrammes, puisque ceux-ci sont des distributions discrétisées. Néanmoins, d'autres distances ont montré leur supériorité pour la comparaison d'histogrammes dans les systèmes de recherche d'images, comme la distance d'*Earth Mover* (Rubner et al., 1998). Les auteurs y présentent aussi la divergence de Jeffrey d_λ pour la comparaison d'histogrammes qui est une extension de la distance de Kullback-Leibler. Cette distance est symétrique dans le cas où $\lambda = 1/2$ et est appelée la divergence de Jensen-Shannon (Fuglede & Topsoe, 2004). Cette dernière mesure est robuste au bruit et numériquement stable. De plus Puzicha et al. (1997) démontrent que la divergence de Jensen-Shannon donne de meilleurs résultats pour la discrimination de texture. Elle est définie pour deux distributions p_X et q_X par :

$$d_\lambda(p_X, q_X) = \lambda d_{KL}(p_X \mid \lambda p_X + (1-\lambda) q_X) + (1-\lambda) d_{KL}(q_X \mid \lambda p_X + (1-\lambda) q_X)$$

Dans le cadre de la théorie de l'information, ces distances mesurent des quantités d'information. La divergence de Kullback-Leibler mesure le nombre de bits supplémentaires nécessaires pour coder le signal en utilisant $p_X(X \mid \theta_2)$ au lieu de $p_X(X \mid \theta_1)$. D'autre part, la divergence de Jensen-Shannon peut être vue comme le nombre de bits supplémentaires nécessaires pour décoder un signal qui est généré par un mélange de deux distributions connues p_X et q_X.

Enfin, ces dernières années ont vu l'apparition de distances fondées uniquement sur les longueurs de code. Li et al. (2004) proposent une distance fondée sur l'information commune que partagent deux objets. Dans cette approche, les objets peuvent être comparés directement sans passer par une phase d'extraction d'information. Une vision englobante de toutes ces distances qui tirent avantage de la compression est donnée par Sculley & Brodley (2006). Ils définissent l'opérateur $C(x)$ qui calcule la longueur de code minimale nécessaire à représenter l'objet x. Cette longueur peut être estimée à l'aide d'encodeurs sans perte. Alors la forme générale des distances est :

$$d_f(x, y) = 1 - \frac{C(x) + C(y) - C(xy)}{f(x, y)}$$

où $f(x, y)$ calcule la similarité entre les deux objets x et y à partir de l'opérateur $C(.)$. La Figure 2.8 présente le schéma qui permet de calculer la distance d_f. La Distance de Compression Normalisée (DCN) (Li et al., 2004) est obtenue pour $f(x, y) = \max\{C(x), C(y)\}$. La Mesure de Dissimilitude fondée sur la Compression (MDC) introduite par Keogh et al. (2004) est obtenue en prenant $f(x, y) = C(x) + C(y)$. La Métrique de Chen-Li (MCL) (Chen et al., 1999) est obtenue en prenant $f(x, y) = C(xy)$. Enfin, la mesure de Cosinus fondée sur la compression (CosS) (Sculley & Brodley, 2006) est définie en prenant $f(x, y) = \sqrt{C(x)C(y)}$. Toute ces mesures ont été définies dans des contextes différents, mais sont unifiées par l'utilisation de la compression $C(.)$ et par leur forme générique. Ces similarités donnent de très bons résultats en classification non supervisée, que ce soit

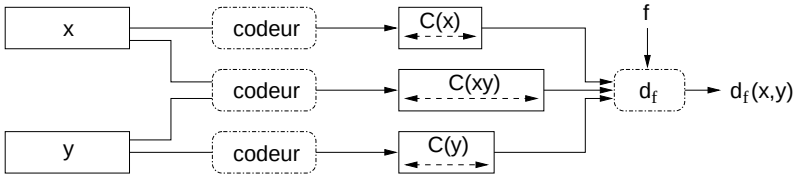


FIG. 2.8 – Schéma qui permet de calculer la distance d_f à l'aide d'un codeur sans perte. Les longueurs de code $C(x)$, $C(y)$ et $C(xy)$ sont utilisées avec la fonction f pour calculer la distance informationnelle d_f entre les deux objets x et y.

pour du texte, l'ADN où des séquences d'images (Keogh et al., 2004). Elles sont d'ailleurs comparées entre elles avec plusieurs modes de compression, où MCL est la meilleure dans le cas traité.
Toutes les mesures de similarité précitées sont non-paramétriques. Donc le choix d'une de ces similarités détermine les performances de l'indexation induite. Comme, nous l'avons fait remarquer le choix de la distance dépend fortement des primitives extraites excepté pour les distances fondée sur la compression.

2.6 L'extraction d'information et l'indexation vues par le prisme de la compression

Il est souligné dans le chapitre §2.4, que la compression est importante pour le stockage des données et que l'extraction d'information peut se faire sans à avoir à décompresser les données. Après une analyse plus approfondie de certaines techniques d'extraction et d'indexation, nous dressons le parallèle qui existe entre la compression et l'extraction d'information objective. Nous rappelons ici le paradigme de la fouille d'information vu comme un canal de communication, où l'extraction d'information objective est un problème de compression (Datcu et al., 2003).

2.6.1 Décorrelation et indépendance statistiques

L'espace des transformées (cf. §2.2.2,§2.4.2) s'avére efficace pour la compression des images avec pertes et pour la caractérisation des textures. Ces transformées sont intéressantes puisqu'elles décorrèlent l'information spatiale et plus précisement décorrèlent les pixels. Ainsi, si les coefficients de la transformée suivent une loi gaussienne et sont décorrélés, ceux-ci deviennent statistiquement indépendants et peuvent être codés indépendemant sans perdre en efficacité de compression. Cependant, les transformées employées ne décorrelent pas totalement les pixels mais essaient de s'en approcher.
La décomposition de Kahrunen-Loéve (Duda et al., 2001) permet de décorréler les échantillons d'un signal. C'est même la transformation qui donne la meilleure décorrélation possible. Par conséquent, coder indépendamment les coefficients de la transformée donne les meilleurs taux de compression. Cependant, cette transformation est peu utilisée en compression puisqu'elle dépend fortement du signal et donc doit aussi être codée lors de la compression. Dans la même famille de transformée linéaire, nous trouvons l'analyse en composantes indépendantes et par pousuite de projections.
Les transformées en ondelettes et les TCD sont beaucoup utilisées en compression avec pertes des images. Wornell (1993) démontre que les ondelettes orthonormales ont un pou-

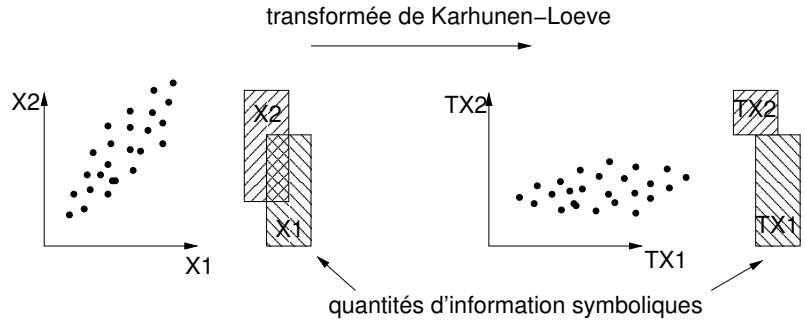


FIG. 2.9 – La décorrelation des variables X_1 et X_2 par la transformée de Karhunen-Loeve permet de partitioner l'information. X_1 et X_2 ont de l'information qui se recoupe, tandis que TX_1 et TX_2, résultats de la transformée, n'ont pas d'information commune. Par exemple, si X_2 est supprimé, la moitié de l'information est perdue. Au contraire, après la transformation, l'information est concentrée sur une des variables. Par exemple, si TX_2 est supprimé, alors la perte d'information est fortement réduite.

voir de décorrelation sur les processus fractals dont la densité spectrale de puissance suit une décroissance $1/f$. Il montre d'autre part qu'une décomposition en ondelettes de coefficients décorrélés génére par transformation inverse des processus fractal $1/f$. Comme les images présentent ce type de décroissance spectrale, ces décompositions sont bien adaptées pour décorréler les pixels des images.
Toutes ces transformées de décorrélation sont intéressantes pour la compression avec pertes, puisqu'elle permettent de séparer les parties des signaux à forte et faible variabilité ou les parties contenant beaucoup et peu d'information. Ainsi, il est naturel de supprimer les parties contenant peu d'information. La Figure 2.9 présente schématiquement comment l'information portée par chaque composante peut varier aprés une transformée de Karhunen-Loéve. Par conséquent, les transformées de décorrélation sont adaptées à l'extraction d'information, puisqu'elles éliminent les redondances et partitionent l'information par le lien qui existe entre l'indépendance statistique et la décorrelation. Une fois l'information partitionée en morceaux d'information indépendants, l'extraction de l'information pertinente est plus aisée.

2.6.2 Modèles générateurs et Modulation Différentielle d'Impulsions Codées

Les modèles générateurs sont des modèles statistiques paramétrés, tels que si X est une variable aléatoire, ces réalisations x sont modélisées par une probabilité $p_{X|\theta}(X = x \mid \theta)$ où θ sont les paramètres. Ce sont ces mêmes paramètres qui servent comme primitives pour caractériser ces réalisations. Ainsi, l'extraction d'information correspond à estimer les paramètres qui modélisent au mieux les réalisations. Lors de la caractérisation des textures par des champs aléatoires de Gauss-Markov (cf. §2.2.2), les primitives extraites correspondent aux paramètres estimés. Dans de nombreux cas, il est associé à la probabilité conditionnelle une fonction de prédiction $g(\theta)$, telle que la probabilité ne dépend que de la différence $e = x - g(\theta)$. Cette différence est appelée erreur résiduelle et suit une distribution $p_E(E = e)$ connue a priori. Alors, l'egalité $p_{X|\theta}(X = x \mid \theta) = p_E(E = x - g(\theta))$ est obtenue. De cette égalité, un estimateur $\hat{\theta}(x)$ des paramètres est calculé en suivant un principe de longueur minimale de codage (cf. §3.2).
Dans de bonnes conditions, ce processus d'extraction d'information est aussi un codeur d'information fondé sur la MDIC. Le principe est de coder l'erreur résiduelle suite à la

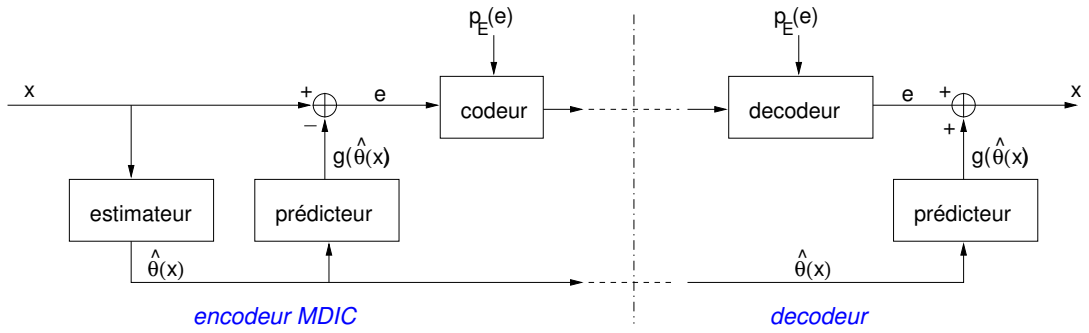


FIG. 2.10 – Schéma d'un codeur par Modulation Différentielle d'Impulsions Codées. Le schéma est fondé sur une prédiction statistique du signal dont l'erreur résiduelle est codée.

prédiction. Un schéma possible de MDIC est présenté dans la Figure 2.10. Ce schéma de codage permet de réaliser une compression sans perte des réalizations x quand la fonction de prédiction et la probabilité d'erreur sont bien choisies. Ce mode de compression est une manière de partitioner l'information en deux. Ainsi, l'information est porté d'un coté par les paramètres et de l'autre par l'erreur résiduelle. Aussi lors de l'extraction d'information, seuls les paramètres sont pris en considération, c'est-à-dire qu'une partie de l'information est extraite. En analyse de texture par des champs aléatoires de Gibbs-Markov (Bader et al., 1995; Zalesny et al., 2002), cette approche se comprend d'autant plus. En effet, à partir de paramètres, ils est possible de générer ou de prédire des textures visuellement semblables. En résumé, la modèlisation par modèles générateurs correspond à la création d'un codeur d'information.

2.6.3 Indexation et quantification vectorielle

Comme nous l'avons vu au §2.5.2, le groupage non-supervisé permet la création d'un index. Par exemple, McLean (1993) utilise la quantification vectorielle pour grouper des textures. Cela vient du fait que les méthodes classiques de clustering ont leur équivalent en quantification vectorielle. La quantification vectorielle, dont l'état de l'art est résumé dans (Gray & Neuhoff, 1998), est utilisée pour la compression avec pertes. Ce type de processus se place dans le cadre de la théorie débit-distortion (Shannon, 1948) qui analyse le compromis entre les pertes et la quantité d'information encodée. Ce processus de compression est encore une fois intéressant pour l'extraction d'information, puisqu'il permet de quantifier la quantité d'information tout en quantifiant les pertes. D'ailleurs, dans la plupart des cas, la similarité utilisée pour confronter deux objets sert aussi comme moyen de mesurer les pertes d'information qu'il y a entre les deux objets. Si la quantité de pertes autorisée pour avoir une bonne discrimination est bien choisie, alors la quantité d'information extraite peut être déterminée par le compromis débit-distorsion.
En conséquence, les deux domaines de clustering et de quantification vectorielle ne forment qu'un seul monde. D'ailleurs l'algorithme Lloyd-Max et le k-moyennes sont exactement identiques malgrés leur découverte dans deux domaines séparés.

2.7 Concept pour l'extraction d'information

En raison, des observations faites dans les chapitres précédents, il parait clair que le cadre de la théorie de l'information et de la compression est recommandé pour la

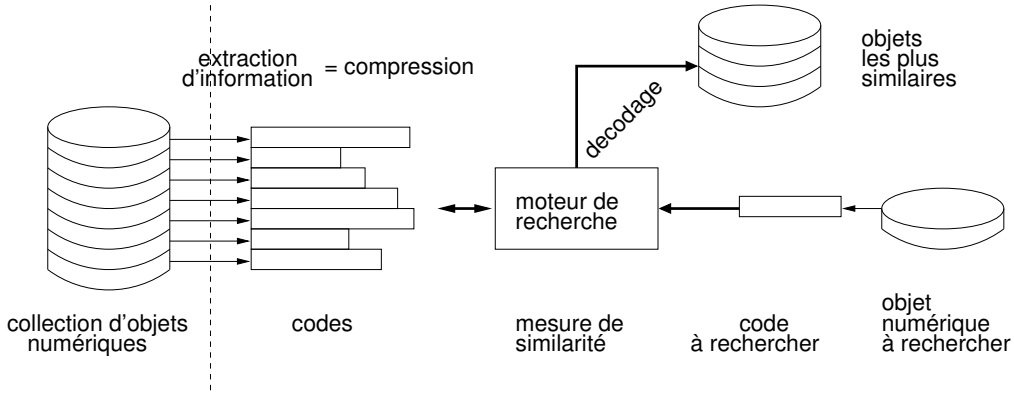


FIG. 2.11 – La compression est vue comme un moyen d'extraire l'information sous forme de code, que le moteur de recherche utilise pour comparer les objets. En effet ces codes contiennent la même information que celle contenue dans les objets. Lors d'une requête, les codes les plus similaires sont séléctionnés et décodés pour obtenir les objets les plus similaires à l'exemple.

conceptualisation d'un système de recherche par le contenu. Durant les étapes clés d'extraction d'information, de groupage et de comparaison, la compression intervient de manière implicite ou explicite. Pour ces raisons, nous voulons faire intervenir explicitement les méthodes de compression pour créer un processus d'extraction d'information non-supervisé. Nous pensons que cette démarche permet de soustraire l'influence du concepteur du système de recherche, qui apporte son information a priori sur les données. Par exemple, untel préfére utiliser les primitives de textures, tandis que l'autre préfére utiliser la géomètrie pour créer leur système RIBC. Ce genre d'approche amène toujours une spécialisation des modèles des données. Et en l'occurence, c'est ce que nous cherchons à éviter par une approche fondée sur la théorie de l'information et la compression. Dans ce travail, nous ne clamons pas arriver à nous abstraire totalement de ce problème, mais nous essayons de faire en sorte que le système de recherche soit le plus indépendant possible de l'application résultante. Ce que nous voulons dire par là, c'est que par exemple une application qui cherche uniquement des ronds dans les images n'a pas intérêt à utiliser des descripteurs de couleurs, et l'application qui se focalise sur les couleurs n'a pas intérêt à être fonder sur les primitives de formes. Cependant, nous voulons obtenir une description commune qui satisfasse au mieux toutes les attentes.
D'autres part dans la plupart des sytèmes d'extraction d'information, l'information est stockée en marge des données (cf. Figures 2.1, 2.5), ce qui constitue une redondance de l'information dans la base de données. Alors par la compression, nous souhaitons obtenir une représentation des données qui soit compacte et qui supporte une système de recherche par le contenu (cf. Figure 2.6). Ainsi, aucune information suplémentaire n'a besoin d'être stockée contrairement aux autres approches. Un tel système d'extraction de données est esquissé dans la Figure 2.11. Ce schéma reste simple, mais souligne le fait que l'information est contenue dans le code, et se suffit à elle-même pour discriminer les objets.
Enfin, dans le paradigme de la fouille d'information, un moteur de recherche peut être vue philosophiquement comme un canal de communication entre une base de données contenant B bits d'information et plusieurs utilisateurs pouvant traité au maximum U bits d'information où $B \gg U$. Le résultat d'un requête est comme une compression avec pertes de la base de données, où seule une petite partie de l'information est trans-

mise. Ainsi, la compression est nécessaire pour faciliter au mieux le passage de l'information dans le goulot d'étranglement engendré par la capacité du canal lié à l'utilisateur. De plus, comme l'extraction d'information ne doit pas être liée à un utilisateur ou une application, nous ne pouvons pas considérer un codage source-canal pour extraire l'information. D'aprés Shannon, un codage source suivi d'un codage canal est optimal pour la transmission d'information. Par conséquent, seul un codage source ou une compression est envisageable pour extraire l'information qui transitera par plusieurs canaux dépendants des utilisateurs ou même des requêtes de recherche.

Chapitre 3

Fondements théoriques

Nous établissons, dans ce chapitre, le cadre théorique dans lequel nous évoluerons pour formaliser l'extraction d'information. Nous passerons continument des modèles de données statistiques à la compression et au codage en passant par l'estimation de paramètres. Nous présentons aussi les deux courants de la théorie de l'information de Shannon et de Kolmogorov en y inluant leurs liens. Tout au long de ce chapitre, nous nous efforcerons de faire le rapprochement avec l'extraction d'information dans les parties opportunes.

3.1 Les modèles de données

La modélisation stochastique consiste à extraire des données des informations utiles. Le point de vue particulier des statistiques est de considérer les données comme la réalisation d'une expérience aléatoire à laquelle correspond un modèle mathématique. Le modèle mathématique est vu alors comme une loi statistique décrivant la possibilité d'apparition des données. Les méthodes d'inférence des modèles sont au coeur de la modélisation, puisqu'elles permettent de contrôler la pertinence des modèles vis à vis des données en se basant sur des critères objectifs. La modélisation statistique intervient quand il n'est pas possible de décrire des données sûrement et que des tendances suffisent à la compréhension du phénomène sous-jacent.
La notion de modèle statistique peut être formalisé comme une famille de distributions de probabilité définies sur l'espace des observations correspondant aux données. Si X est la variable aléatoire, alors ces réalisations notées x appartiennent à l'espace des observations $\mathcal{X}$. L'inférence de modèle consiste à choisir une distribution de probabilité dans la famille à partir des données et la notion de processus stochastique est nécessaire à l'inférence de ces modèles.
Pour la suite, les variables aléatoires seront notées en majuscules et leurs réalisations en minuscules. D'autre part, la notation $p(X) = p_X(X = .)$ fera référence à la fonction de densité de probabilité ou à la probabilité si l'espace $\mathcal{X}$ est continu ou discret. $p(x) = p_X(X = x)$ représentera la valeur de $p(X)$ prise en x.

$$p(X) : \mathcal{X} \rightarrow [0, 1] \tag{3.1}$$

$$x \rightarrow p_X(X = x) \tag{3.2}$$

L'inférence de la loi consiste à calculer la fonction $p(X)$. Comme il est précisé plus haut,

un modèle statistique est une famille de probabilités. Quand cela est possible, cette famille est décrite par un ensemble de paramètres θ qui donnent un cadre pratique pour manipuler ces lois. Ces distributions seront notées $p(X \mid \theta)$ qui correspondent en fait à des lois conditionnelles. Un des problèmes rencontrés avec la paramétrisation est l'identifiabilité, c'est-à-dire qu'il n'existe pas deux paramètres θ_1 et θ_2 tels que $p(X \mid \theta_1) = p(X \mid \theta_2)$. Si ce problème n'a pas lieu d'être, alors inférer le modèle consiste à choisir ou à estimer le meilleur paramètre selon certains critères objectifs. Dans le cadre bayésien, ces lois représentent la probabilité d'apparition d'un événement x relativement à une connaissance a priori θ où la parametrisation est la réalisation d'une variable aléatoire Θ. Par la loi de Bayes, une probabilité conditionnelle est définie par l'équation suivante :

$$p(x \mid \theta) = \frac{p(x, \theta)}{p(\theta)} \tag{3.3}$$

A titre d'information, il est aussi possible de définir d'autres familles de probabilité, telles que les modèles semi paramétriques ou non paramétriques. Dans les sections suivantes, nous décrivons les processus stochastiques et nous nous intéressons aux critères qui permettent de choisir les probabilités et les familles de probabilités.

3.1.1 Les processus stochastiques

Un processus stochastique $X = \{X_t : t \in \mathcal{T}\}$ est une suite de variables aléatoires indexées et prenant leurs valeurs dans un unique espace des observations $\mathcal{X}$. Le processus X est lui même une variable aléatoire. Il est courant de considérer un processus à temps discret où l'ensemble $\mathcal{T}$ est à valeurs discrètes. Le cas continu correspond à une indexation continue du processus. Un processus stochastique est complétement déterminé par la connaissance de la distribution de probabilité conjointe sur un espace discret $\mathcal{T}$. Dans la plupart des cas l'espace discret correspond à une indexation des variables aléatoires sur l'ensemble d'entiers $\mathbb{N}$. Prenons un espace d'indexation $\mathcal{T}$ discret et fini. Celui-ci peut être mis en bijection avec l'intervalle $[0, n]$ où n est le cardinal de l'ensemble. Alors le processus se réécrit par $X = \{X_i : 1 \leq i \leq n\}$. Nous nous référons à ce type de processus par la notation X^n pour mettre en avant le cardinal de l'index.
Les processus stochastiques peuvent être classés en sous-classes en particularisant leur distribution de probabilité. Nous trouvons les processus indépendants et identiquement distribués (i.i.d.), les processus markoviens, ergodiques et stationnaires. Nous présentons par la suite les deux premières catégories.

3.1.2 Les processus indépendants et identiquement distribués

Premièrement un processus stochastique est indépendant si sa loi conjointe suit la condition d'indépendance statistiquesuivante :

$$p(X) = p(\{X_t : t \in \mathcal{T}) = \prod_{t \in \mathcal{T}} p(X_i) \tag{3.4}$$

Deuxièmement, le processus est identiquement distribué si toutes les variables aléatoires composant le processus suivent la même distribution. Cette condition se traduit par :

$$\forall (t, u) \in \mathcal{T}^2 \quad p(X_t) = p(X_u) \tag{3.5}$$

Finalement, un processus i.i.d. vérifie les deux conditions présentées. Les processus i.i.d. seront à la base de l'inférence de modèle.

3.1.3 Les processus markoviens

Les processus i.i.d. constituent une modélisation restrictive et les variables du processus peuvent être dépendantes. Les processus de Markov permettent de modéliser cet aspect. Soit un processus de Markov $X^n = \{X_1, ...X_n\}$ composé de plusieurs variables aléatoires. Un processus de Markov d'ordre 1 est caractérisé par la relation suivante concernant l'indépendance des variables :

$$\forall n \in \mathbb{N} \quad p(X_n \mid X_{n-1}...X_1) = p(X_n \mid X_{n-1}) \tag{3.6}$$

La variable X_n dépend uniquement de la variable précédente. Cette relation markovienne peut être facilement généralisée aux ordres supérieurs, tel que la variables X_n dépend uniquement des d variables précédentes où d est l'ordre du processus de Markov. En appliquant la loi de Bayes et la relation markovienne, la probabilité globale du processus se calcule alors par la relation suivante :

$$p(X^n) = p(X_1) \prod_{k=2}^{n} p(X_k \mid X_{k-1}) \tag{3.7}$$

Plus généralement, la markovianité est appréhendée par l'utilisation d'une fonction S qui prend ses valeurs dans un espace de dimension finie. Par exemple, pour définir une processus markovien d'ordre d, prenons cette fonction telle que :

$$\forall n > d \quad S(X^{n-1}) = \{X_{n-d}, ...X_{n-1}\} \tag{3.8}$$

Alors la relation de markovianité se traduit par les deux équations suivantes :

$$p(X_n \mid X^{n-1}) = p(X_n \mid S(X^{n-1})) \tag{3.9}$$

$$\forall n > d \quad p(X^n) = p(X^d) \prod_{k=d+1}^{n} p(X_k \mid S(X^{k-1}) \tag{3.10}$$

La fonction S donnée reste simple, mais dans certains cas celle-ci peut devenir plus complexe. Supposons pour l'instant que cette fonction nous est donnée et soit bien définie. Alors, cette modélisation nous permet de prédire le futur à partir des réalisations passées par le biais de la relation markovienne (Eq.3.9). Pour cela, une fonction de prédiction g est utilisée telle que la prédiction correspond à définir :

$$\forall n \in \mathbb{N} \quad X_n = g(S(X^{n-1})) + E_n \tag{3.11}$$

où E^n est un processus stochastique. De plus, si nous faisons l'hypothèse que la connaissance de X_n et $S(X^{n-1})$ permet de déterminer sûrement $S(X^n)$, alors $E^n \mid g$ est un processus stochastique indépendant. En rajoutant l'hypothèse que le prossecus X^n est conditionellement identiquement distribué, le processus $E^n \mid g$ devient i.i.d. En d'autres termes, par le biais de ces hypothèses, nous ramenons n'importe quel processus markovien à un processus stochastique paramétré par g et i.i.d. Aussi, les méthodes d'estimation de paramètres présentées dans les chapitres suivants pourront s'appliquer au processus markovien en considérant cette transformation.

3.1.4 Les champs aléatoires de Gibbs-Markov

Les processus présentés précédemment ne sont pas adaptés à la modélisation de signaux multidimensionnels. En effet, les processus markoviens modélisent des signaux unidimensionnels. Alors, la famille des champs aléatoires de Gibbs-Markov intervient pour la modélisation de signaux tels que les images ou les vidéos. Cette modélisation permet de prendre en compte les dépendances d'une variable avec ses voisins.
Supposons, que nous avons une espace fini multidimensionnel Ω. A chaque position $s \in \Omega$ est associée une variable aléatoire X_s. Le champs aléatoire X défini sur Ω est la collection des variables $\{X_s : s \in \Omega\}$.
Un tel champ aléatoire est markovien quand il vérifie la propriété suivante :

$$\forall s \in \Omega \quad p(X_s \mid X_t : t \in \Omega, t \neq s) = p(X_s \mid X_t : t \in N_s) \tag{3.12}$$

Cette propriété veut dire que la variable X_s dépend uniquement des variables qui font partie du voisinage N_s de la position s. Ce voisinage ne contient pas le site s et est indépendant des sites.
Les champs aléatoires de Gibbs sont eux aussi définis à partir d'un système de voisinage. Un voisinage N est autorisé si $t \in N_s \Leftrightarrow s \in N_t$ et $s \notin N_s$. Un système de voisinage induit un système de cliques $\mathcal{C} = \{c_k\}$ tel que chaque clique est un sous-ensemble de Ω. Ce sous-ensemble vérifie le fait que deux éléments lui appartenant sont voisins par N (Winkler, 1995). Les interactions locales, c'est-à-dire des interactions qui agissent entre des éléments d'une clique, peuvent maintenant servir à exprimer des énergies et des potentiels. La fonction d'énergie d'une réalisation x du champ aléatoire s'exprime comme une somme de potentiels V_c associés à chaque clique :

$$U(x \mid \theta) = \sum_{c \in \mathcal{C}} V_c(x \mid \theta) \tag{3.13}$$

Ce terme d'énergie est influencé par la forme des cliques et les potentiels de celles-ci, ce qui résulte en une multitude de dépendances statistiques. De plus la fonction d'énergie d'une variable seule s'exprime comme la somme de toutes les énergies des cliques qui incluent la variable :

$$U_s(x_s \mid \theta) = \sum_{c \in \mathcal{C}, s \in c} V_c(x_s \mid \theta) \tag{3.14}$$

Un champ aléatoire de Gibbs suit alors une distribution de la forme :

$$p(X \mid \theta) = \frac{e^{-U(X|\theta)}}{Z} \tag{3.15}$$

$$Z = \sum_x e^{-U(x|\theta)} \tag{3.16}$$

où Z est la fonction de partition qui permet de normaliser la distribution. Le théorème de Hammersley-Clifford (Spitzer, 1971) démontre qu'il existe une équivalence totale entre les champs aléatoires de Markov et de Gibbs. Il en découle que la vision locale d'un champ de Gibbs se traduit par :

$$p(x_s \mid x_t : t \in N_s, \theta) = \frac{e^{-U(x_s|x_t:t\in N_s,\theta)}}{Z(\theta)} \tag{3.17}$$

$$Z(\theta) = \sum_{x_s, x_t} e^{-U(x_s|x_t:t\in N_s,\theta)} \tag{3.18}$$

où $U(x_s \mid x_t : t \in N_s, \theta)$ est une fonction d'énergie locale. Ainsi, tout champ aléatoire vérifiant une des propriétés est appelé champs de Gibbs-Markov. Cette fonction d'énergie locale permet de modéliser les interactions qu'il existe entre les sites. Néanmoins, reconstruire la probabilité du champs X à partir de la relation markovienne reste difficile. Aussi dans de nombreux cas, la fonction de pseudo-vraisemblance (PV) est considérée et s'exprime par :

$$p_{PV}(X \mid \theta) = \prod_{s \in \Omega} p(X_s \mid X_t : t \in N_s, \theta) \tag{3.19}$$

Considérons, maintenant le champ aléatoire $U = \{U_s : s \in \Omega\}$ tel que une réalisation de U_s sachant θ est donné par :

$$u_s(\theta) = U(x_s \mid x_t : t \in N_s, \theta) \tag{3.20}$$

Alors la probabilité du champ U est donnée par l'Eq. 3.21 en considérant la fonction de pseudo-vraisemblance :

$$p_{PV}(U \mid \theta) = \prod_{s \in \Omega} \frac{e^{-U_s(\theta)}}{Z(\theta)} = \prod_{s \in \Omega} p(U_s \mid \theta) \tag{3.21}$$

Nous notons que considérer la fonction de pseudo-vraisemblance revient à considérer un processus stochastique sous-jacent de variables i.i.d. En effet, si plus aucune relation n'existe entre les variables U_s du champs aléatoire, celui-ci est réduit à une processus i.i.d. en réordonnant l'ensemble Ω sous forme d'un espace unidimensionnel. Ainsi par cette approximation, nous pouvons nous rapporter aux méthodes d'estimation utilisées sur les processus stochastiques de variables i.i.d.

3.1.5 Exemples de champs aléatoires

Nous donnons ici la définition de deux champs aléatoires de Gibbs-Markov qui seront utilisés par la suite pour la modélisation de signaux définis sur une grille d'échantillonnage à trois dimensions. Nous présentons d'abord les champs de Gauss-Markov, puis les champs aléatoires autobinomiaux.

3.1.5.1 Les champs aléatoires de Gauss-Markov

Ces champs aléatoires ont été introduits par Chellappa & Kashyap (1983) pour la modélisation de champs bidimensionnels. Nous appliquons cette modélisation aux signaux à trois dimensions. Tout d'abord nous considérons une grille d'échantillonnage $\Omega = \{s = (i, j, t), s \in \Omega\}$ telle que la grille est un pavé de dimensions $l_i \times l_j \times l_t$.
Pour définir la fonction d'énergie, nous considérons les voisinages symétriques présentés dans la Figure 3.1 et qui vérifient la condition de Gibbs. De plus à chaque type de voisinage N, on peut associer un demi voisinage $N^{1/2}$ vu que N est symétrique. Nous indexons les pixels du demi voisinage par un ensemble $\{r \in N^{1/2}\}$. Si le pixel central est au point $(0, 0, 0)$, alors les pixels appartenant au voisinage N sont donnés par l'union $\{r\} \cup \{-r\}$. Par cette définition, le voisinage N_s d'un site $s = (i, j, t)$ correspond alors à l'ensemble $\{s + r : r \in N^{1/2}\} \cup \{s - r : r \in N^{1/2}\}$. La fonction d'énergie d'un champ de Gauss-Markov est définie par :

$$U(x_s \mid x_t : t \in N_s, \theta) = -\frac{1}{2} \frac{\| x_s - \sum_{r \in N^{1/2}} \theta_r \frac{x_{s+r} + x_{s-r}}{2} \|_2^2}{\sigma^2} \tag{3.22}$$

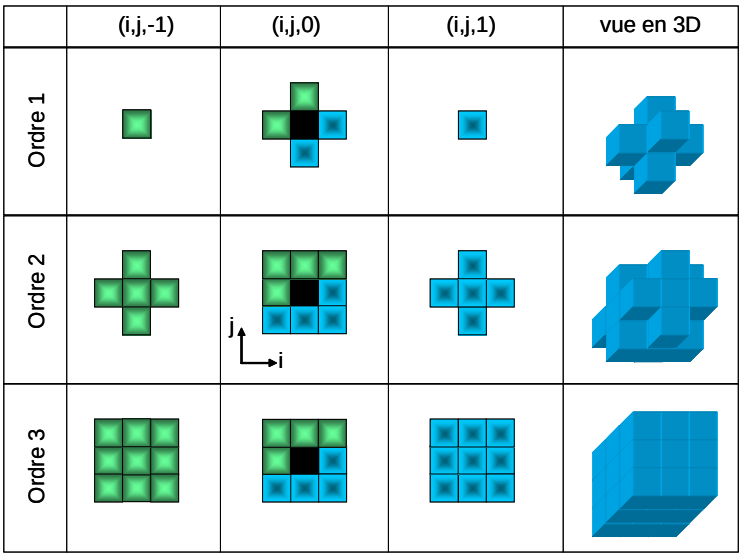


FIG. 3.1 – Des voisinages symétriques à trois dimensions sont présentés. Le pixel central est coloré en noir. Nous voyons que les voisinages sont séparés en deux tel que les pixels r appartenant à $N^{1/2}$ sont verts et leurs opposés sont bleus. De plus, $N^{1/2}$ est un voisinage causal dans ce cas. On peut distinguer des voisinages des 3 premiers ordres.

où chaque θ_r est un paramètre scalaire associé à chaque position r du demi voisinage. Ainsi le vecteur de paramètres θ correspond à l'ensemble $\{\sigma, \theta_r : r \in N^{1/2}\}$. Nous réécrivons la fonction d'énergie en introduisant une variable d'erreur e_s :

$$x_s = \sum_{r \in N^{1/2}} \theta_r \frac{x_{s+r} + x_{s-r}}{2} + e_s \tag{3.23}$$

Alors en combinant cette définition à Eq. 3.17, $p(E_s \mid \theta)$ devient une loi gaussienne de moyenne nulle et de variance σ. Si de plus nous considérons la fonction de pseudo-vraisemblance du champ X, alors la distribution du champ $E \mid \theta$ associée aux variables E_s est donné par :

$$p(E \mid \theta) = \prod_{s \in \Omega} p(E_s \mid \theta) \tag{3.24}$$

Cela revient à considérer que $E \mid \theta$ est un processus stochastique i.i.d. où les variables $E_s \mid \theta$ suivent la loi $\mathcal{N}(0, \sigma)$. En fait cette modélisation, vise à prédire un pixel central par une combinaison linéaire de ses voisins. Etant donné que des combinaisons linéaires sont utilisées, cette modélisation admet une représentation linéaire (Chellappa & Kashyap, 1985). Nous définissons un index linéaire de Ω qui correspond à une fonction bijective $a : \Omega \mapsto [1, 2, ..., l_i l_j l_t]$. Un exemple d'index est présenté dans la Figure 3.2. De même nous définissons un index linéaire de $N^{1/2}$ qui correspond à la fonction bijective $b : N^{1/2} \mapsto [1, ...|N^{1/2}|]$. Par ces index, les réalisations x, e sont transformées en deux vecteurs $\vec{x}, \vec{e}$ tels que $\vec{x}_{a(s)} = x_s$ et $\vec{e}_{a(s)} = e_s$, respectivement. D'autre part, nous considérons le vecteur de paramètres $\vec{\theta}$ obtenu par l'égalité $\vec{\theta}_{b(r)} = \theta_r$. Enfin, nous introduisons la matrice G de taille $l_i l_j l_t \times \mid N^{1/2} \mid$ définie par ses éléments :

$$G_{a(s),b(r)} = \frac{x_{s+r} + x_{s-r}}{2} \tag{3.25}$$

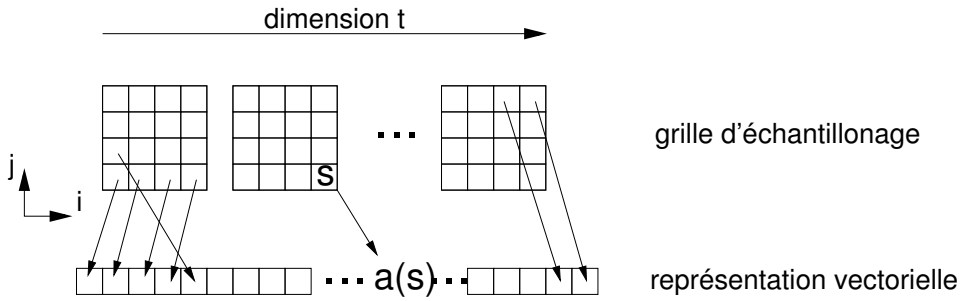


FIG. 3.2 – La grille d'échantillonnage Ω est indexée par la fonction $a(s)$. Les carrés représentent la grille sur l'espace (i, j) qui évolue selon la troisième dimension t. Cette indexation résulte en une représentation vectorielle de la grille.

En fin de compte la modélisation se réécrit sous forme linéaire à partir de l'Eq. 3.23 par l'équation suivante :

$$\vec{x} = G\vec{\theta} + \vec{e} \tag{3.26}$$

où $\vec{e}$ est une réalisation d'un processus gaussien de moyenne nulle et de variance σ. Enfin, Bader et al. (1995) montrent la puissance de cette modélisation pour la construction de texture à partir d'un vecteur de paramètres en utilisant un échantillonneur de Gibbs ou une reconstruction directe à partir d'un vecteur d'erreurs généré aléatoirement.

3.1.5.2 Les champs aléatoires autobinomiaux

Comme précédemment, nous considérons le même type de voisinage et de grille d'échantillonnage. Les champs aléatoires autobinomiaux ont été présentés originellement par Cross & Jain (1983b) et ont ensuite été utilisé pour la modélisation des images satellitaires dans (Schroder et al., 1998). La fonction d'énergie unitaire permettant la définition d'un champ autobinomial est exprimée par :

$$U(x_s \mid x_t : t \in N_s, \theta) = -\log \begin{pmatrix} \mathcal{G} \\ x_s \end{pmatrix} - x_s \eta \tag{3.27}$$

$$\eta = \theta_0 + \sum_{r \in N} \theta_r \frac{x_{s+r} + x_{s-r}}{\mathcal{G}} \tag{3.28}$$

où $\mathcal{G}$ est la valeur maximale que X_s peut prendre. Contrairement à la modélisation de Gauss-Markov, l'espace des observations des variables X_s est limité à l'intervalle d'entiers $[0, 1, ..., \mathcal{G}]$. De plus $\begin{pmatrix} n \\ k \end{pmatrix} = \frac{n!}{k!(n-k)!}$.
Avec cette fonction d'énergie, la distribution de Markov s'écrit sous la forme multinomiale suivante :

$$p(x_s \mid x_t : t \in N_s, \theta) = \begin{pmatrix} \mathcal{G} \\ x_s \end{pmatrix} q^{x_s} (1-q)^{\mathcal{G}-x_s} \tag{3.29}$$

$$q = \frac{1}{1 + e^{-\eta}} \tag{3.30}$$

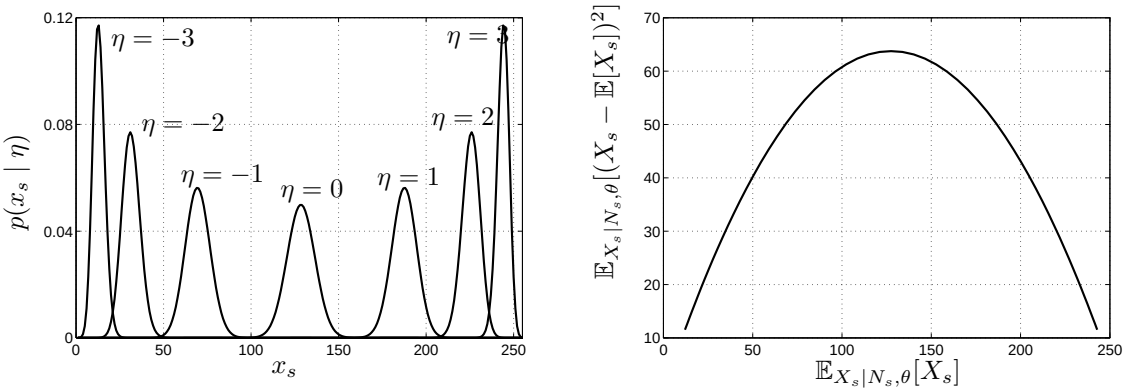


FIG. 3.3 – A droite, nous voyons la distribution $p(X_s \mid \eta)$ quand $\mathcal{G} = 255$ et pour différentes valeurs de η. A gauche, la moyenne est représentée contre la variance.

De ces définitions, la moyenne et la variance conditionnelles se calculent simplement par :

$$\mathbb{E}_{X_s|N_s,\theta}[X_s] = \frac{\mathcal{G}}{1+e^{-\eta}} \tag{3.31}$$

$$\mathbb{E}_{X_s|N_s,\theta}[(X_s - \mathbb{E}[X_s])^2] = \mathcal{G}\frac{e^{-\eta}}{(1+e^{-\eta})^2} \tag{3.32}$$

Certaines courbes caractérisant les propriétés de cette modélisation sont exposées dans la Figure 3.3. Nous remarquons tout d'abord que la variance est contrainte par la moyenne. En effet, si la moyenne est proche de $\mathcal{G}/2$ alors la variance est d'autant plus élevée. Au contraire sur les bords de l'intervalle $[0, \mathcal{G}]$, les variances sont plus petites. Dans la formation des images, il y a plus de chances d'avoir des valeurs autour de $\mathcal{G}/2$ que sur les bords, ce qui produit de plus grandes variances au centre. Encore une fois, le pixel central est prédit par une fonction non linéaire de ses voisins et des paramètres θ. Nous verrons qu'il est possible d'approximer les interactions non linéaires par une modélisation linéaire semblable à celle décrite au §3.1.5.1.

3.2 Estimation de paramètres

3.2.1 Estimateurs bayésiens

L'estimation de paramètres consiste à calculer, à partir des observations, les paramètres qui détermineront la probabilité qui colle au mieux aux données. Nous considérons une variable aléatoire X. L'estimation de paramètre est en fait une fonction de l'espace des observations vers l'espace des paramètres et est notée $\hat{\theta}(x)$. Le but final de l'estimation est que la probabilité $p(x \mid \hat{\theta}(x))$ modélise *bien* la réalisation x. Dans le cadre bayésien, le paramètre est vu comme une variable aléatoire dont la distribution doit être évaluée.
Pour déterminer les meilleures fonctions d'estimation, un critère objectif est de minimiser le risque de Bayes lié à une fonction de coût $c(\theta, \hat{\theta})$. Ce risque de Bayes est défini comme un risque moyen exprimé par :

$$R = E[c(\Theta, \hat{\theta}(X))] = \int\int c(\theta, \hat{\theta}(x))p(x,\theta)dxd\theta \tag{3.33}$$

Certaines définitions de la fonction de coût, que nous allons décrire par la suite, nous donnent quelques estimateurs de référence. Tout d'abord, la fonction de coût quadratique $c_q(\theta, \hat{\theta}) = (\theta - \hat{\theta})^T(\theta - \hat{\theta})$ donne l'estimateur du Minimum de l'Erreur Quadratique Moyenne (MEQM). Le risque moyen associé s'écrit comme suit en utilisant la définition des probabilités conditionnelles (Eq. 3.3) :

$$R_q = \int p(x) \int (\theta - \hat{\theta})^T(\theta - \hat{\theta}) p(\theta \mid x) dx d\theta \tag{3.34}$$

Le minimum de ce risque moyen est atteint pour la fonction d'estimation MEQM exprimée par :

$$\hat{\theta}_{MEQM}(x) = \int \theta p(\theta \mid x) d\theta \tag{3.35}$$

C'est la moyenne des paramètres connaissant la loi de probabilité $p(\theta \mid x)$ a posteriori. Une autre fonction de coût est le risque uniforme défini par :

$$c_u(\theta, \hat{\theta}) = \begin{cases} 0 & \text{, si } \mid \theta - \hat{\theta} \mid < \delta/2 \\ 1 & \text{, si } \mid \theta - \hat{\theta} \mid \geq \delta/2 \end{cases} \tag{3.36}$$

Alors le minimum du risque moyen associé R_u est atteint avec l'estimateur du Maximum A Posteriori (MAP) décrit par :

$$\hat{\theta}_{MAP}(x) = \arg\max_{\theta} p(\theta \mid x) \tag{3.37}$$

A titre de première comparaison, si le mode de la distribution a posteriori est égal à la moyenne alors les estimateurs MEQM et MAP sont équivalents. Néanmoins, ces deux estimateurs font appel à la distribution a posteriori qui dépend de la distribution a priori des paramètres $p(\Theta)$ et qui n'est pas forcément connue.
Toutefois, en se restreignant à un paramètre considéré comme non aléatoire, la distribution a priori $p(\Theta)$ est un dirac et le risque moyen de Bayes n'est plus adapté. Alors la fonction de vraisemblance $p(x \mid \theta)$ intervient pour définir l'estimateur du Maximum de Vraisemblance (MV) :

$$\hat{\theta}_{MV}(x) = \arg\max_{\theta} p(x \mid \theta) \tag{3.38}$$

Cet estimateur est encore appelé estimateur de la log-vraisemblance où la fonction à minimiser est $-\log p(x \mid \theta)$. Nous notons que l'estimateur MV est égal à l'estimateur MAP pour une distribution des paramètres a priori uniforme et est égal à l'estimateur MEQM pour une distribution a priori symétrique. Bien que l'estimateur MV ne soit pas fondé sur le risque de Bayes, il est qualifié de bayésien.
L'approche minimax est aussi utilisée pour l'estimation de paramètres et est issue de la théorie des jeux. Ce critère s'écrit

$$\min_{\hat{\theta}(X)} \max_{\theta} R(\theta, \hat{\theta}(X)) \tag{3.39}$$

Cette procédure minimise le risque maximum encouru où le risque est défini par :

$$R(\theta, \hat{\theta}(X)) = \int p(x \mid \theta) c(\theta, \hat{\theta}(x)) dx \tag{3.40}$$

Ce critère très prudent a pour objectif de donner la protection maximale contre la pire situation. Contrairement au risque de Bayes où celui-ci est moyenné sur l'ensemble des

paramètres, le minimax prend le pire cas.
Ces estimateurs sont couramment utilisés pour la modélisation de processus stochastiques et nous serviront par la suite. Toutefois, nous avons souligné le problème lié à la distribution a priori des paramètres qui est discuté dans §3.2.4. Tout d'abord, nous rappelons les limites de ces estimateurs.

3.2.2 Limites des estimateurs

Un estimateur peut être vu comme une statistique, qui est une fonction définie sur l'espace des observations dans un autre espace. Pour définir les limites, nous restreignons la statistique $T(X)$ à prendre ses valeurs dans un espace de dimension 1 et à prendre les paramètres θ de dimension d. Alors, la moyenne de l'estimateur est donnée par :

$$\psi(\theta) = \mathbb{E}_\theta[T(X)] = \int T(x)p(x \mid \theta)dx \tag{3.41}$$

Sous certaines conditions d'échange entre intégrale et dérivée, on peut toujours écrire :

$$\mathbb{E}_\theta[\nabla_\theta \log p(x \mid \theta)] = 0 \tag{3.42}$$

Supposons que u soit un vecteur de dimension d, alors nous pouvons écrire :

$$\mathbb{V}ar_\theta[T(X)] \geq \nabla_\theta \psi(\theta)^T I(\theta)^{-1} \nabla_\theta \psi(\theta) \tag{3.43}$$

où $I(\theta)$ est la matrice d'information de Fisher qui admet la définitions suivante :

$$I(\theta) = \left\{ \mathbb{E}_\theta \left[\frac{\partial \log p(X \mid \theta)}{\partial \theta_i} . \frac{\partial \log p(X \mid \theta)}{\partial \theta_j} \right] \right\}_{i,j \leq d} = \{I_{i,j}(\theta)\}_{i,j \leq d} \tag{3.44}$$

Cette matrice d'information représente les variations des entropies locales avec les variations du paramètre θ. L'inégalité Eq. 3.43 est extensible au cas où la statistique $T(X)$ prend ses valeurs dans un espace de dimension d. L'inégalité est légèrement modifiée, telle que la matrice de covariance de $T(X)$ devienne supérieure à la matrice de Fisher :

$$\mathbb{C}ov_\theta[T(X)] \geq \nabla_\theta \psi(\theta)^T I(\theta)^{-1} \nabla_\theta \psi(\theta) \tag{3.45}$$

Cette inégalité entre deux matrices M et N définies semi positives et de taille $d \times d$ se traduit par :

$$M \geq N \quad \text{si} \quad \forall x \in \mathbb{R}^d \quad x^T M x \geq x^T N x \tag{3.46}$$

Si nous prenons $T(X)$ comme un estimateur de paramètre $\hat{\theta}(X)$, tel que celui-ci est sans biais, c'est-à-dire que $\psi(\theta) = \theta$, alors nous avons l'égalité suivante $\nabla_\theta \psi(\theta) = Id$ qui donne la borne de Cramer-Rao :

$$\mathbb{C}ov_\theta[\hat{\theta}(X)] \geq I(\theta)^{-1} \tag{3.47}$$

Cette borne indique que n'importe quelle fonction d'estimation admet un variation des estimés supérieure à la borne de Cramer-Rao.
Un processus stochastique $X = \{X_1, X_2, ..., X_n\}$ où les variables sont i.i.d est une variable aléatoire. En conséquence, toutes les procédures d'estimation et les limites s'y appliquent.

En particulier, sa matrice d'information de Fisher est égale à n fois la matrice de Fisher unitaire :

$$I(\theta) = nI_1(\theta) = n\left\{\mathbb{E}_\theta\left[\frac{\partial \log p(X_1 \mid \theta)}{\partial \theta_i}.\frac{\partial \log p(X_1 \mid \theta)}{\partial \theta_j}\right]\right\}_{i,j\leq d} \tag{3.48}$$

Ainsi la borne de Cramer-Rao se réécrit :

$$\mathbb{C}ov_\theta[\hat{\theta}(X)] \geq \frac{I_1(\theta)^{-1}}{n} \tag{3.49}$$

On remarque donc que cette limite se réduit quand le nombre de variables n grandit. Ainsi lors de l'estimation de paramètres, il vaut mieux considérer une variable aléatoire comme un processus stochastique de variables i.i.d. pour gagner sur la limite d'estimation et fournir de meilleurs estimés. Cela se comprend par le fait que considérer n réalisations indépendamment conduit à n estimés. Au contraire, considérer les n réalisations comme un processus stochastique conduit à un estimé qui est bien meilleur. A titre indicatif, cette limite est atteinte pour la famille des lois gaussiennes.

3.2.3 Estimation des paramètres en deux niveaux d'inférence

Les méthodes d'estimation précitées sont optimales quand la famille de modèles paramétriques est connue et fixée. Considérons maintenant le cas où nous avons plusieurs familles de modèles paramétriques pour modéliser un processus stochastique X (MacKay, 1991). Nous notons cet ensemble de familles de modèles $\{\mathcal{M}_1, ...\mathcal{M}_n\}$ sur lequel la variable aléatoire $\mathcal{M}$ prend ses valeurs. De plus, à chaque modèle $\mathcal{M}_i$ est associé un vecteur aléatoire de paramètres Θ_i. Sur cet ensemble de variables, il est possible de définir une variable aléatoire Θ dont l'espace des observations est l'union de tous les espaces des observations des Θ_i. De plus, on suppose que le processus X et le modèle $\mathcal{M}$ sont indépendants connaissant les paramètres, ce qui se traduit par :

$$p(X, \mathcal{M} \mid \Theta) = p(X \mid \Theta)p(\mathcal{M} \mid \Theta) \tag{3.50}$$

Le problème posé est d'estimer le meilleur modèle et les meilleurs paramètres pour représenter une réalisation du processus stochastique. Deux estimations doivent donc être faites.
Premièrement, l'inférence des paramètres peut être faite par un estimateur MAP connaissant le modèle et les données. La relation nécessaire au calcul du MAP est obtenue en combinant la relation de Bayes (Eq. 3.3) et l'équation (Eq. 3.50) :

$$p(\Theta \mid X, \mathcal{M}) = \frac{p(X \mid \Theta, \mathcal{M})p(\Theta \mid \mathcal{M})}{p(X \mid \mathcal{M})} \tag{3.51}$$

A partir du moment où les réalisations de $\mathcal{M}$ et X sont connues et notées $\mathcal{M}_i$ et x, la variable Θ est restreinte à la variable Θ_i et l'estimateur MAP correspondant est :

$$\hat{\theta}(x, \mathcal{M}_i) = \hat{\theta}_i(x) = \arg\max_{\theta_i} p(x \mid \theta_i, \mathcal{M}_i)p(\theta_i \mid \mathcal{M}_i) \tag{3.52}$$

Deuxièmement, la meilleure famille de modèles doit aussi être estimée. L'estimateur MAP est alors utilisé en maximisant la relation de Bayes suivante, où $p(\mathcal{M})$ est la distribution a priori des familles :

$$p(\mathcal{M} \mid X) = \frac{p(X \mid \mathcal{M})p(\mathcal{M})}{p(x)} \propto p(X \mid \mathcal{M})p(\mathcal{M}) \tag{3.53}$$

Du fait des indépendances supposées, le terme $p(X \mid \mathcal{M})$, encore appelé évidence du modèle, peut être calculé par marginalisation pour chaque réalisation $\mathcal{M}_i$. La marginalisation s'opère sur la probabilité conjointe des réalisations et des paramètres. Elle s'écrit :

$$p(X \mid \mathcal{M}_i) = \int p(X \mid \theta_i, \mathcal{M}_i) p(\theta_i \mid \mathcal{M}_i) d\theta_i \tag{3.54}$$

Notons que cette intégrale ne peut se calculer analytiquement que pour certains modèles, et que son calcul numérique est souvent très complexe. Toutefois, nous verrons par la suite que des approximations permettent son évaluation. D'autre part nous verrons les principes qui permettent de déterminer les probabilités a priori, par des considérations apparues dans le domaine de la théorie de l'information. Ainsi, pour inférer le meilleur modèle décrivant les données, deux problèmes se posent. Il faut pouvoir calculer l'évidence du modèle et sa loi a priori.

3.2.4 Sélection de modèle

Les modèles et leurs paramètres ne peuvent pas être sélectionnés ensemble par le calcul direct de leurs vraisemblances. En effet, plus une modèle est complexe, plus il a des chances de s'ajuster aux données. Partant de cette constatation, il est clair que pour un jeu fini de réalisations des données, une sélection de la famille de modèles et des paramètres obtenus par la maximisation de la vraisemblance, aboutira inévitablement à la surestimation de la complexité du modèle. Aussi pour pallier cet effet, le principe du rasoir d'Occam énonce que des modèles simples représentent mieux les données que des modèles complexes. D'ailleurs, MacKay (2003) montre qu'un facteur de complexité du modèle apparaît dans le calcul de l'évidence du modèle. Il constate que la distribution $p(\Theta_i \mid x, \mathcal{M}_i)$ présente un pic important au niveau de l'estimateur MAP $\hat{\theta}_i$. En conséquence, cette distribution a posteriori est approximée par une gaussienne centrée sur $\hat{\theta}_i$ et est développée au second ordre par les séries de Taylor. Ainsi l'évidence est approximée par :

$$\begin{aligned} p(x \mid \mathcal{M}_i) &\approx p(x \mid \hat{\theta}_i, \mathcal{M}_i) \times p(\hat{\theta}_i \mid \mathcal{M}_i) det(\frac{H}{2\pi})^{-1/2} \\ \text{Evidence} &\approx \text{Vraisemblance} \times \text{Facteur d'Occam} \end{aligned} \tag{3.55}$$

où $H = -\nabla^2 \log p(\hat{\theta}_i \mid x, \mathcal{M}_i)$ est le laplacien évalué à l'estimé MAP. Il est montré, que pour un modèle paramétrique régulier possédant k degrés de liberté et un processus possédant n réalisations, le logarithme du facteur d'Occam est égal à :

$$\log p(\hat{\theta}_i \mid \mathcal{M}_i) - \frac{k}{2} \log (\frac{n}{2\pi}) - \log det(I(\hat{\theta}_i))^{-1/2} \tag{3.56}$$

où I est la matrice d'information de Fisher définie à l'Eq. 3.44. Ces approximations permettent le calcul de l'évidence et par la même occasion permettent l'inférence du modèle par MV. Toutefois, si la probabilité a priori des paramètres n'est pas calculable ou inconnue, cette procédure est inutilisable. Jeffreys (1948) introduit une loi a priori non informative. Cette idée vient d'une invariance par changement de variables. Soit g un difféomorphisme de l'espace des paramètres, alors le changement de variables donne l'égalité suivante sur les distributions :

$$p(\theta) = \mid det(\frac{dg}{d\theta}) \mid p(g(\theta)) \tag{3.57}$$

La distribution a priori de Jeffreys définie dans l'Eq. 3.58 vérifie cette égalité. Elle est donc utilisée du fait de cette invariance. Cette distribution est dite non informative puisque elle n'apporte pas d'information sur le modèle utilisé. En effet, un autre modèle peut être défini par le changement de variable avec une distribution a priori qui satisfait le changement.

$$p_J(\Theta_i \mid \mathcal{M}_i) = \frac{det(I(\Theta_i))^{-1/2}}{\int det(I(\theta))^{-1/2} d\theta} \tag{3.58}$$

Ainsi, en considérant cette distribution a priori, le logarithme du facteur d'Occam pour des modèles paramétriques devient :

$$-\frac{k}{2}\log(\frac{n}{2\pi}) - \log\int det(I(\theta))^{-1/2} d\theta \tag{3.59}$$

Nous avons vu que dans quelques cas et sous certains a priori, le problème de l'estimation des paramètres et la sélection de famille de modèles peut être résolu. Seulement, à ce stade il n'y a pas de moyen de définir la distribution a priori des familles de modèles. Nous verrons dans la suite que des principes issus de la théorie de l'information permettent de régler ce problème. D'autres part, nous remarquons que le calcul de l'évidence du modèle nécessite l'estimation des paramètres. Ainsi, la procédure d'estimation implique que pour chaque modèle, leurs paramètres doivent être estimés dans un premier temps, puis le modèle le plus vraisemblable peut être sélectionné en omettant la distribution a priori des modèles. On parle ici de sélection de modèle, puisque les évidences sont calculées avant d'être comparées pour prendre le meilleur modèle. L'estimation de paramètres est différente puisque les estimés sont généralement obtenus par calcul direct. En conséquence, si on a un grand nombre de familles de modèles, la sélection peut s'avérer être complexe.
Nous présentons maintenant le moyen de calculer l'évidence pour des modèles linéaires (O Ruanaidh & Fitzgerald, 1996) définis par l'Eq. 3.60. Nous présentons ce type de modélisation puisqu'il est très répandu en traitement du signal.

$$x = G\theta + e \tag{3.60}$$

où e est la réalisation d'un processus i.i.d gaussien centré $\mathcal{N}(0,.)$ et de variance σ inconnue. G est une transformée linéaire agissant sur une vecteur de paramètres θ. L'estimation des paramètres, première étape d'inférence par Maximum de Vraisemblance (Eq. 3.52), nous donne :

$$\hat{\theta} = (GG^T)^{-1}G^T x \tag{3.61}$$
$$\hat{\sigma}^2 = x^T x - (G\hat{\theta})^T(G\hat{\theta}) \tag{3.62}$$

L'évidence de ce modèle linéaire suivant les approximations (Eq. 3.55) et l' a priori précédent (Eq. 3.58) est donnée par :

$$p(x|G,\mathcal{N}(0,.)) \approx \frac{\pi^{-n/2}\Gamma(\frac{q}{2})\Gamma(\frac{n-q}{2})|G^TG|^{-1/2}}{4R_\delta R_\sigma(\hat{\theta}^T\hat{\theta})^{q/2}\hat{\sigma}^{n-q}} \tag{3.63}$$

où n est la taille de x, q la taille de θ et $R_\delta R_\sigma$ est une constante de normalisation. Le deuxième niveau d'inférence (Eq. 3.53) pour la sélection du modèle peut être réalisé en admettant une distribution uniforme sur l'espace des familles de modèles.

Nous avons vu jusqu'a présent plusieurs procédures pour la modélisation de processus stochastique en s'appuyant sur des méthodes d'estimation. Nous présentons dans la suite en quoi ces estimations sont équivalentes à mesurer l'information et compresser les processus stochastiques.

3.3 Compression sans perte et modélisation

Dans cette section, nous rappelons les définitions portant sur les mesures de l'information introduites par Shannon (1948). Dans un premier temps, nous présentons l'entropie de Shannon, la divergence de Kullback-Leibler et l'information mutuelle comme mesures de l'information. Dans un second temps, nous présentons comment ces mesures peuvent être utilisées pour compresser les données. Enfin, nous soulignons le lien qu'il existe entre l'estimation bayésienne et la compression.

3.3.1 Mesure de l'information

Shannon (1948) définit l'entropie d'une variable aléatoire X prenant ses valeurs dans un espace des observations discret $\mathcal{X}$, comme une mesure de l'incertitude qu'il existe sur les réalisations de la variable. L'entropie (Eq. 3.64) est en fait la moyenne de la variable $-\log p(X)$ qui mesure l'incertitude d'obtenir X. En effet, plus $p(X)$ est petit, plus cette mesure est grande et l'événement est incertain :

$$H(X) = -\sum_{x\in\mathcal{X}} p(x)\log p(x) \tag{3.64}$$

Nous notons que l'entropie est maximale pour une distribution uniforme sur $\mathcal{X}$ et qu'elle est nulle pour un distribution de dirac. En effet, la probabilité de dirac représente une réalisation certaine, n'apportant aucune information. Cette mesure s'exprime en bits d'information car la base logarithmique considérée est 2.
A partir de cette définition, Shannon énonce son premier théorème de codage de source ou de compression sans perte. Il montre, que si on associe un mot de code c_x à chaque réalisation x de l'espace des observations, encore appelé dictionnaire, tel que ce code garantisse un décodage unique, alors la longueur moyenne des mots de code est supérieure à l'entropie. La longueur des mots de code $L(c_x)$ se mesure en bits de Turing, c'est-à-dire qu'elle prend une valeur entière.

$$\sum_{x\in\mathcal{X}} p(x)L(c_x) \geq H(X) \tag{3.65}$$

De plus, il montre que l'égalité est atteinte pour un code qui vérifie $-\log p(x) = L(c_x)$ et qui est appelé code de Shannon-Fano.
D'autre part, Shannon montre qu'il est possible de construire un codage préfixe binaire vérifiant l'inégalité de Kraft (Eq. 3.66) et qui permet de coder la source X.

$$\sum_{x\in\mathcal{X}} 2^{-L(c_x)} \leq 1 \tag{3.66}$$

Ce codage vérifie $L(c_x) = \lceil -\log p(x) \rceil$ et la longueur moyenne de codage reste inférieure à $H(X)+1$. Par ce théorème, Shannon montre qu'il est possible de coder une source tout en s'approchant de la limite établie par l'entropie. Néanmoins, ce théorème s'applique pour une source dont les réalisations se succèdent à l'infini. D'autre part, il faut noter, que cette limite dépend fortement de la connaissance de la distribution des données.

3.3.2 La divergence de Kullback-Leibler et l'information mutuelle

La divergence de Kullback-Leibler est introduite et utilisée pour définir le principe de discrimination minimum (Kullback & Leibler, 1951). Ce principe consiste à choisir une distribution $p(X)$ lorsqu'un a priori $q(X)$ est disponible et que $p(X)$ doit satisfaire un jeu de contraintes. Ce principe de discrimination minimum est encore interprété comme le principe du maximum d'entropie (Jaynes, 1968). La divergence de Kullback-Leibler est définie par :

$$d_{KL}(p \mid q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \tag{3.67}$$

Dans le cadre de la compression, cette divergence peut se réinterpréter en tant qu'une redondance de codage. Cette quantité d'information correspond à coder une source X qui suit la probabilité $p(X)$ en utilisant un code de Shannon-Fano construit à partir de $q(X)$. En l'occurrence, si on se trompe sur le modèle statistique de X, alors la quantité d'information supplémentaire utilisée lors du codage sera supérieure à la divergence de Kullback-Leibler. De plus, nous notons que cette quantité est toujours positive et est nulle si et seulement si $p(X) = q(X)$. Enfin, notons que cette divergence n'est pas symétrique.
Maintenant, considérons deux variables aléatoires X et Y telles que $p(X, Y)$ soit la distribution conjointe. Alors l'information mutuelle (Shannon, 1948) entre ces deux variables est définie par :

$$I(X, Y) = d_{KL}(p(X, Y) \mid p(X)p(Y)) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{3.68}$$

Cette quantité d'information correspond à la quantité d'information supplémentaire nécessaire au codage des deux sources lorsque les variables sont considérées comme étant indépendantes. L'information mutuelle est souvent interprétée comme la quantité d'information partagée par les deux variables. Cette quantité admet aussi les définitions suivantes (Eq. 3.69). Dans ce cas, cette mesure s'interprète comme la différence entre la longueur minimale de codage de X (ou Y) seul et la longueur de codage de X sachant Y (ou $Y \mid X$).

$$I(X, Y) = H(X) + H(Y) - H(X, Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \tag{3.69}$$

Cette quantité est positive, symétrique et s'annule si et seulement si les variables sont indépendantes statistiquement.
Une multitude d'autres mesures de l'information existent (Taneja, 2001). Mais, nous utilisons les mesures précédentes puisqu'elles permettent de déduire des bornes pour la compression.

3.3.3 Codage et estimation

Dans ce paragraphe, nous montrons en quoi l'estimation de paramètres est un schéma de codage de source. Plus particulièrement, l'estimation MAP est un schéma de codage d'un processus stochastique X^n en deux parties. Ce processus stochastique est composé de n variables i.i.d associées une distribution inconnue $p(X)$. La limite de codage de Shannon du processus X^n est égale à $nH(X)$.

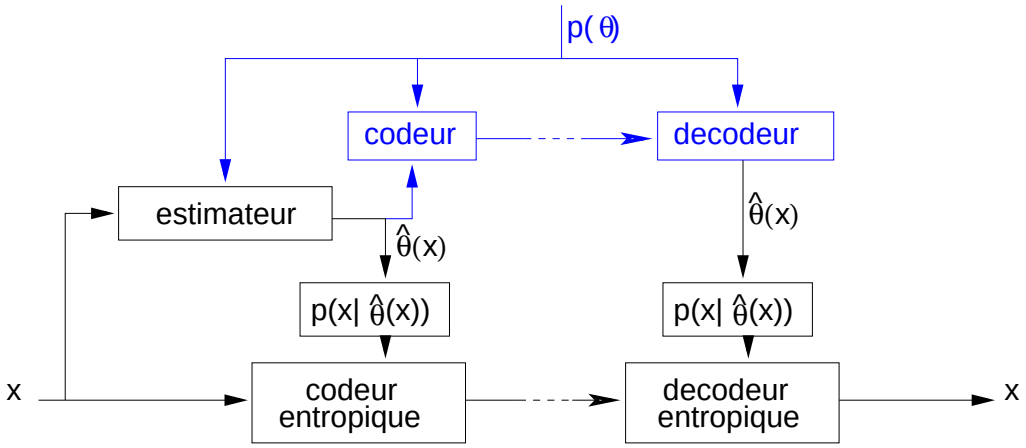


FIG. 3.4 – Cette figure présente comment l'estimation de paramètre peut être vue comme un problème de codage d'information. La partie en noire correspond à une estimation MV où les paramètres sont connus au codage et décodage. Le modèle paramétrique $p(X \mid \Theta)$ est connu par le codeur et le décodeur. La partie bleue ajoutée, correspond à une estimation MAP où la distribution a priori des paramètres est connue du codeur et du décodeur. Ce processus de codage est optimal en longueur de code puisque l'estimation vise à la minimiser.

Nous rappelons l'estimateur MAP sous sa forme logarithmique :

$$\hat{\theta}_{MAP}(x^n) = \arg\min_{\theta} \{-\log p(x^n \mid \theta) - \log p(\theta)\} \tag{3.70}$$

$$= \arg\min_{\theta} \left\{ -\sum_{k=1}^{n} \log p(x_k \mid \theta) - \log p(\theta) \right\} \tag{3.71}$$

Cette estimation minimise la longueur de code nécessaire à encoder conjointement la réalisation x^n et l'estimé $\hat{\theta}_{MAP}$. Imaginons, un codeur et un décodeur qui ont accès à la distribution a priori $p(\Theta)$, alors l'estimé peut être codé avec un code de Shannon-Fano. En l'occurrence, le vecteur de paramètres $\hat{\theta}_{MAP}$ est encodé avec $-\log p(\hat{\theta}_{MAP})$ bits. D'un autre coté, chaque réalisation x_k peut être encodée avec $-\log p(x_k \mid \hat{\theta}_{MAP})$ bits. Le schéma de ce type de codage est présenté dans la Figure 3.4. Nous notons que l'augmentation de la longueur du processus diminue l'entropie moyenne liée aux variables indépendantes. En effet, quand n tend vers l'infini, la longueur de code moyenne tend vers $\frac{1}{n}(H(X^n \mid \hat{\theta}_{MAP}) - \log p(\theta)) = H(X \mid \hat{\theta}_{MAP}) - \frac{\log p(\theta))}{n}$. Ainsi, la longueur de code moyenne liée aux paramètres tend vers zéro. Comme nous l'avons remarqué au §3.2.2, l'augmentation de n est préférable pour de meilleurs estimations. Nous concluons que les deux points de vue s'accordent. Contrairement, au codage entropique, ce codage est optimisé pour une réalisation du processus. Si maintenant, nous voulons coder une infinité de réalisations du processus, nous voulons comparer la longueur obtenue à l'entropie du signal. Considérons donc la divergence de Kullback-Leibler entre la vraie distribution $q = p(X^n)$ et la distribution $r = p(X^n \mid \hat{\theta}(X^n))p(\hat{\theta}(X^n))$:

$$d_{KL}(q \mid r) = \sum_{x^n} p(x^n) \log \frac{p(x^n)}{p(x^n \mid \hat{\theta}(x^n))p(\hat{\theta}(x^n))} = -\sum_{x^n} p(x^n) \log p(\hat{\theta}(x^n) \mid x^n) \tag{3.72}$$

Par la construction du schéma d'encodage, nous sommes sûrs que la distribution $p(\Theta \mid X^n)$ est déterministe ce qui implique que la divergence $d_{KL}(q \mid r)$ est nulle. En conséquence,

ce système de codage admet comme limite inférieure de longueur de code, l'entropie de Shannon. Toutefois, nous avons supposé pour déduire Eq. 3.72 que la distribution a priori des paramètres vérifie la relation de Bayes. Considérons, que les paramètres soient encodés avec une distribution a priori $w(\theta)$ qui ne vérifie pas la relation de Bayes. Alors, la divergence devient :

$$d_{KL}(q \mid r) = \sum_{x^n} p(x^n) \log \frac{p(\hat{\theta}(x^n))}{w(\hat{\theta}(x^n))} \tag{3.73}$$

$$= \sum_{\theta} \sum_{x^n : \hat{\theta}(x^n)=\theta} p(x^n) \log \frac{p(\theta)}{w(\theta)}$$

$$= \sum_{\theta} p(\theta) \log \frac{p(\theta)}{w(\theta)} \tag{3.74}$$

Ainsi la redondance de codage équivaut à la divergence de Kullback-Leibler entre la vraie distribution des paramètres $p(\Theta)$ et l'a priori donnée $w(\Theta)$. Ce que nous appelons la vraie distribution correspond à la distribution induite par la relation de Bayes connaissant la vraie distribution $p(X^n)$. En conséquence, le système de codage est optimal au sens du codage de Shannon-Fano si la distribution a priori est bien choisie puisque le codage atteint la limite de Shannon. Néanmoins, dans bien des cas la distribution des données $p(X^n)$ n'est pas connue ce qui ne permet pas de reconstituer l'a priori sur les paramètres. Cependant, nous savons que l'information supplémentaire à encoder vient du choix de l'a priori sur les paramètres. Aussi, nous avons un critère objectif pour le choix du meilleur a priori donné par l'Eq. 3.74. Il est démontré que la minimisation par MAP décrite dans l'Eq. 3.70 tend à réduire la redondance de codage des paramètres, en minimisant la divergence de Kullback-Leibler. Nous concluons que l'estimation bayésienne MAP est équivalente à construire un schéma de codage sans perte qui tend vers la limite de codage de Shannon.
Nous avons vu comment l'estimation de paramètres peut se réénoncer en termes d'efficacité de codage. Cependant, cette minimisation se fait dans le cadre d'une famille de modèles fixée. Nous décrivons par la suite comment le principe de longueur minimale de codage sert pour l'estimation de famille de modèles.

3.3.4 Principe de Longueur de Description Minimale (LDM)

Comme nous l'avons vu précédemment, la distribution a priori est déterminante pour le codage et l'estimation. Rissanen (1983) présente le principe de Longueur de Description Minimale (LDM) pour le codage universel. Pour un certain jeu de réalisations et une collection finie de modèles, le principe LDM sélectionne le modèle engendrant la longueur minimale de description des données. La validité du principe LDM dépend des propriétés de la longueur de description employée ou plus précisément des propriétés du schéma de codage sous-jacent. Pour formaliser ces propriétés, Rissanen (1986) détermine la borne inférieure de la redondance de codage. Puis, Clarke & Barron (1990) détermine la borne inférieure de la redondance minimax du code avec lequel les données sont encodées pour une classe de modèles donnée. Ainsi les schémas de codage, aboutissant à des longueurs de description dont la redondance vérifie ces bornes inférieures, sont appelés des codes universels.
Nous nous plaçons dans le contexte d'une classe de modèles paramétriques définie par

$\mathcal{M} = \{p(X^n \mid \theta), \theta \in \mathbb{R}^k\}$. Soit un processus X^n qui est modélisé par une des distributions de la famille. La redondance de codage est exprimée par la divergence de Kullback-Leibler telle que $R(\theta, q) = d_{KL}(p(X^n \mid \theta) \mid q(X^n))$. Le problème posé est de déterminer la longueur minimax de redondance définie par :

$$\min_{q(X^n)} \max_{\theta} R(\theta, q) \tag{3.75}$$

Ce critère consiste à trouver la distribution $q(X^n)$ qui induit la longueur de code la plus proche du pire cas de codage. A l'inverse du cas décrit au §3.3.3, ici nous supposons que le processus est distribué selon une des lois conditionnelles de $\mathcal{M}$. Ne connaissant pas cette loi, nous voulons trouver une distribution qui codera au mieux les réalisations.
La première borne de la redondance de codage est obtenue par Rissanen (1986). Il suppose qu'un estimateur $\hat{\theta}$ se calcule avec une précision de $1/\sqrt{n}$. Il montre que pour toute distribution $q(X^n)$, excepté certaines distributions appartenant à un ensemble qui tend vers l'ensemble vide à l'infini, l'inégalité suivante est vérifiée presque sûrement :

$$\lim_{n\to\infty} R(\hat{\theta}, q) \geq \frac{k}{2} \log n \tag{3.76}$$

Cette inégalité sert à définir la limite d'un code universel. Si nous voulons construire un codeur universel du processus, nous voulons calculer la meilleure distribution q qui induit la longueur de code minimale. Mais cette inégalité réécrite dans Eq. 3.77 montre que dans la majorité des cas une distribution q ne fera pas mieux que le codeur MAP présenté au §3.3.3 en prenant une distribution a priori telle que $-\log p(\theta) = \frac{k}{2} \log n$. Cet a priori uniforme implique par conséquent que l'estimation MAP est une estimation MV.

$$-\sum_{x^n} p(x^n \mid \hat{\theta}) \log q(x^n) \geq -\sum_{x^n} p(x^n \mid \hat{\theta}) \log p(x^n \mid \hat{\theta}) + \frac{k}{2} \log n \tag{3.77}$$

D'autre part, Dawid (1987) particularise l'inégalité précédente en montrant que :

$$-\log q(x^n) \geq -\log p(x^n \mid \hat{\theta}(x^n)) + \frac{k}{2} \log n \tag{3.78}$$

Les distributions q qui ne satisfont pas ces deux inégalités engendrent la classe des codes universels. D'ailleurs, Rissanen (n.d.) montre qu'en prenant la distribution $q(X^n)$ définie dans l'Eq. 3.79, les inégalités s'inversent en inégalités strictes.

$$-\log q(x^n) = -\log p(x^n \mid \hat{\theta}(x^n)) + \frac{k}{2} \log n + C \tag{3.79}$$

où C est un constante de normalisation telle que la distribution vérifie l'inégalité de Kraft (Eq. 3.66). Ce codage universel revient au final à une description de l'information en deux parties comme elle a été présentée dans la Figure 3.4.
De plus, Clarke & Barron (1990) montrent que la borne obtenue pour le problème du minimax (Eq. 3.75) est donnée par :

$$\min_{q(X^n)} \max_{\theta} R(\theta, q) = \log p(\hat{\theta}) - \frac{k}{2} \log (\frac{n}{2\pi}) - \log det(I(\hat{\theta}))^{-1/2} \tag{3.80}$$

Cette borne est encore appelée la complexité d'information stochastique. Dés lors, nous remarquons que cette borne correspond au facteur d'Occam exprimé pour les modèles

paramétriques dans Eq. 3.56. Par conséquent, nous soulignons le fait que les problèmes d'inférence bayésienne sont équivalents aux problèmes de codage de source.
Maintenant que nous avons ces bornes, nous pouvons décrire le principe LDM qui permet de sélectionner la meilleure famille de modèles. En termes de codage, la meilleure famille de modèles paramétriques est celle qui donne un code universel qui minimise la longueur de représentation. Ainsi, l'estimation de paramètres et la sélection de modèles s'exprime dans le cadre du codage comme le principe de longueur de description minimale en s'appuyant sur le code universel (Eq. 3.79) :

$$\min_{\theta,k} \left\{ -\log p(x^n \mid \theta) + \frac{k}{2} \log n \right\} \tag{3.81}$$

D'autre part, en s'appuyant sur la borne minimax et en prenant un a priori de Jeffreys (Eq. 3.58), le principe de longueur de description minimale se généralise par :

$$\min_{\theta,k} \left\{ -\log p(x^n \mid \theta) + \left(\frac{k}{2} \log n + \log \int det(I(\phi))^{-1/2} d\phi \right) \right\} \tag{3.82}$$

Il est à noter que dans toutes ces façons d'aborder le problème, il résulte que le signal est toujours codé en deux parties. C'est-à-dire que l'information est séparée en deux quantités, dont une concerne le codage du modèle et l'autre concerne le signal exprimé dans le modèle. Les inégalités précédentes nous montrent que ne connaissant pas a priori la distribution du processus, le meilleur moyen de le coder est de diviser l'information en deux parties. En fin de compte, la modélisation paramétrique est réinterprétée comme le codage des propriétés du signal et le codage du reste nécessaire à la détermination de celui-ci. Alors, ces deux aspects du signal peuvent être mesurés par les quantités d'information qu'ils véhiculent.

3.4 Compression avec pertes et extraction d'information

Nous nous sommes intéressés à la modélisation et au codage sans perte des processus stochastiques. Nous avons vu que le codage est un moyen de modéliser les données. Dans la théorie du codage, la théorie débit-distorsion s'est intéressée au codage de signaux tel que des pertes soient autorisées pour la reconstruction de ceux-ci. Dans la modélisation bayésienne des signaux cet aspect n'est pas décrit. Aussi, nous présentons cette théorie pour modéliser approximativement des signaux ou des objets numériques. Ensuite, nous présentons la quantification vectorielle comme une méthode d'approximation du signal efficace dans le cadre de cette théorie. Enfin, nous finissons pas décrire comment ces méthodes permettent de modéliser partiellement les données et d'extraire l'information.

3.4.1 Théorie débit-distorsion

Dans cette théorie, nous nous posons le problème de savoir combien de bits sont nécessaires à coder les réalisations d'une variable X sachant que certaines pertes sont autorisées lors du décodage. De plus le modèle $p(X)$ est connu comme dans le théorème de codage de Shannon. Pour formaliser ce problème, considérons une fonction de perte $d(x, \hat{x})$ définie entre une réalisation en entrée x et la réalisation reconstruite $\hat{x}$. Cette fonction permet d'évaluer les pertes engendrées par le processus de codage-décodage. Prenons le cas où nous souhaitons coder l'information avec R bits d'information par

réalisation. Cela est formalisé par l'existence d'un fonction de codage f et une fonction de décodage g codant et décodant n réalisations consécutives du processus :

$$f \quad : \quad \mathcal{X}^n \rightarrow \{1, 2, ...2^{nR}\} \tag{3.83}$$

$$g \quad : \quad \{1, 2, ...2^{nR}\} \rightarrow \hat{\mathcal{X}}^n \tag{3.84}$$

La fonction de perte moyenne D permet d'évaluer la perte d'information occasionnée par ce codage-décodage :

$$D = \mathbb{E}[d(X, \hat{X})] = \frac{1}{n} \sum_{i=1}^{n} d(x_i, g(f(x_i))) \tag{3.85}$$

Cover & Thomas (1991) définissent la région de codage atteignable, comme l'ensemble des couples (R, D) tels qu'il existe une suite de couples codage-décodage (f, g) quand n tend vers l'infini. Ils montrent d'ailleurs que cette région est bornée par la fonction débit-distorsion $R^*(D)$, c'est-à-dire que pour tout couple atteignable (R, D), l'inégalité $R \geq R^*(D)$ est vérifiée. D'autre part, ils démontrent que cette fonction débit-distorsion s'exprime par un problème de minimisation :

$$R^*(D) = \min_{p(\hat{X}|X)} I(X, \hat{X}) \quad \text{s. c.} \quad \mathbb{E}_{X,\hat{X}}[d(X, \hat{X})] \leq D \tag{3.86}$$

Dans cette vision du problème, on cherche à minimiser l'information que X et $\hat{X}$ partagent sous la condition que les pertes n'excédent pas D. La minimisation se fait sur une fonction de codage-décodage statistique $p(\hat{X} \mid X)$ qui représente la probabilité d'obtenir telle sortie sachant telle entrée.
Nous remarquons que dans le cas où il existe une procédure de codage-décodage déterministe $g \circ f$, alors le signal reconstruit représente une partie de l'information de X. En effet, le codage-décodage déterministe correspond à avoir une probabilité d'assignement telle que :

$$p(\hat{x} \mid x) = \begin{cases} 1 & si \quad \hat{x} = g \circ f(x) \\ 0 & sinon \end{cases} \tag{3.87}$$

Alors, l'information mutuelle entre les deux variables se recalcule par :

$$\begin{aligned} I(X, \hat{X}) &= \sum_{\hat{x}} \sum_{x} p(\hat{x} \mid x) p(x) \log \frac{p(\hat{x} \mid x)}{p(\hat{x})} \\ &= \sum_{\hat{x}} \sum_{x, g \circ f(x) = \hat{x}} -p(x) \log p(\hat{x}) \\ &= \sum_{\hat{x}} -\log p(\hat{x}) \sum_{x, g \circ f(x) = \hat{x}} p(x) \\ &= H(\hat{X}) \end{aligned} \tag{3.88}$$

puisque $\sum_{x, g \circ f(x) = \hat{x}} p(x) = p(\hat{x})$. Ainsi, il est clair que dans le cas où l'assignement est déterministe, $\hat{X}$ est une sous-partie de l'information de X car l'information partagée est égale à l'entropie du signal reconstruit.
La fonction débit-distorsion est une fonction décroissante convexe (Cover & Thomas, 1991) et elle présente le compromis qu'il existe entre les pertes et le débit. Plus le débit est bas, plus les pertes sont grandes. Un exemple de fonction débit-distorsion est présenté dans la Figure 3.5.

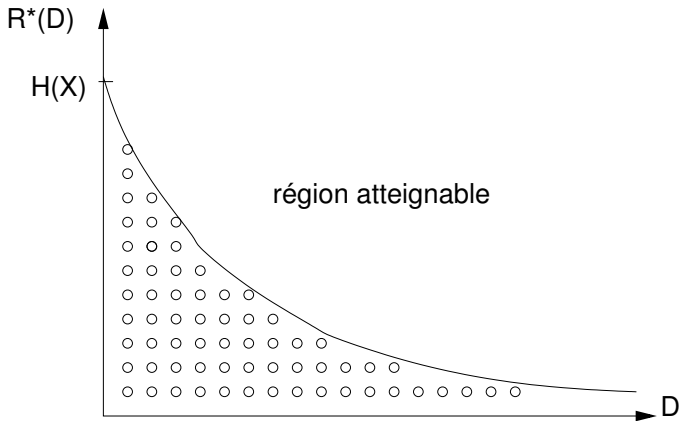


FIG. 3.5 – Cette courbe présente la forme générale d'une courbe débit-distorsion. On peut distinguer la région des couples débit-distorsion atteignables.

3.4.2 Quantification vectorielle et clustering

La quantification vectorielle est le moyen le plus simple d'implémenter la théorie débit-distorsion. En effet, Blahut (1972) montre qu'à partir de l'équation débit-distorsion, il est possible de calculer la fonction de codage-décodage statistique. Nous décrivons dans un premier temps les équations consistantes liées à la fonction débit-distorsion. Puis, dans un second temps nous présentons l'algorithme de l'Espérance-Maximisation qui permet de calculer la fonction itérativement.

3.4.2.1 Equations consistantes

Blahut (1972) montre que l'Eq. 3.86 peut se réécrire sous la forme lagrangienne suivante :

$$\min_{p(\hat{X}|X),\hat{\mathcal{X}}} I(X,\hat{X}) + \beta \mathbb{E}_{X,\hat{X}}[d(X,\hat{X})] + \sum_x \gamma_x \sum_{\hat{x}} p(\hat{x} \mid x) \tag{3.89}$$

où la minimisation se fait sur l'espace de reproduction et la probabilité conditionnelle. Le dernier terme du lagrangien correspond à la contrainte implicite sur l'intégration à un des probabilités conditionnelles. Cette formulation nous permet d'obtenir une paramétrisation de la courbe débit-distorsion qui dépend du paramètre de Lagrange β. Nous noterons par la suite $\mathcal{L}$ le terme à minimiser. Premièrement, Blahut (1972) montre que la dérivation du lagrangien par rapport à la probabilité conditionnelle donne :

$$\frac{\partial \mathcal{L}}{\partial p(\hat{x} \mid x)} = 0 \Rightarrow \log \frac{p(\hat{x} \mid x)}{p(\hat{x})} = -\beta d(x,\hat{x}) - \gamma_x \tag{3.90}$$

Cette équation se réécrit de la manière suivante :

$$p(\hat{x} \mid x) = p(\hat{x}) e^{-\beta d(x,\hat{x}) - \gamma_x} \tag{3.91}$$

Comme γ_x est la constante de normalisation, cette probabilité conditionnelle peut se réécrire :

$$p(\hat{x} \mid x) = \frac{p(\hat{x})}{Z(x,\beta)} e^{-\beta d(x,\hat{x})} \tag{3.92}$$

$$Z(x,\beta) = \sum_{\hat{x}} p(\hat{x}) e^{-\beta d(x,\hat{x})} \tag{3.93}$$

Domaine	$\phi(x)$	$D_\phi(x, \hat{x})$	Divergence
$\mathbb{R}$	x^2	$(x-y)^2$	pertes quadratique
$\{0,1\}$	$x \log x + (1-x) \log (1-x)$	$x \log \frac{x}{y} + (1-x) \log \frac{1-x}{1-y}$	pertes logistique
$\mathbb{R} - \{0\}$	$\mid x \mid$	$\max \{0, -2signe(y)x\}$	pertes de Hinge
$\mathbb{R}^d$	$\|x\|^2$	$\|x-y\|^2$	distance Euclidienne
$\mathbb{R}^d$	$x^T Ax$	$(x-y)^T A(x-y)$	distance de Mahalanobis
d-Simplex	$\sum_{i=1}^d x_i \log x_i$	$\sum_{i=1}^d x_i \log \frac{x_i}{x_j}$	KL-divergence
$\mathbb{R}^+$	$-\log x$	$\frac{x}{y} - \log \frac{x}{y} - 1$	distance de Itakura-Saito

TAB. 3.1 – Quelques divergences de Bregman associées à leur fonction génératrice. La distance Euclidienne, très employée, fait partie de ces divergences tout comme la divergence de Kullback-Leibler.

De plus, par la loi de Bayes, on peut écrire la distribution de $\hat{x}$ en fonction de la probabilité conditionnelle par :

$$p(\hat{x}) = \sum_x p(\hat{x} \mid x) p(x) \tag{3.94}$$

Ces deux équations Eq. 3.92 et Eq. 3.94 constituent les deux premières équations consistantes liées à la courbe débit-distorsion. Nous notons que ces deux équations sont intriquées, ce qui rend leur calcul difficile.
Maintenant, dérivons le lagrangien par rapport à l'espace de reproduction $\hat{\mathcal{X}}$. Le seul terme qui dépend d'un petite variation du mot de code x est la fonction de distance.

$$\frac{\partial \mathcal{L}}{\partial \hat{x}} = 0 \Rightarrow \sum_x p(x, \hat{x}) \frac{\partial d(x, \hat{x})}{\partial \hat{x}} = 0 \tag{3.95}$$

La résolution de cette dernière équation dépend de la fonction de pertes choisie. Néanmoins, Banerjee et al. (2004b) introduisent la classe des divergences de Bregman, pour lesquelles l'équation peut être résolue. Nous présentons d'abord la définition d'une divergence de Bregman. Soit une fonction $\phi : \mathcal{X} \mapsto \mathbb{R}$ qui est strictement convexe et différentiable. De plus, on suppose que $\mathcal{X}$ est un espace vectoriel muni d'un produit scalaire $< ., . >$. Alors, la divergence de Bregman associée à la fonction ϕ est positive et définie par :

$$D_\phi(x, \hat{x}) = \phi(x) - \phi(\hat{x}) - < x - \hat{x}, \nabla \phi(\hat{x}) > \tag{3.96}$$

Alors, Banerjee et al. (2004b) prouvent que la solution de l'Eq. 3.95 ne dépend pas du choix de la divergence et elle correspond à la moyenne conditionnée de l'ensemble des observations :

$$\hat{x} = \mathbb{E}_{X|\hat{x}}[X] = \sum_x p(x \mid \hat{x}) x \tag{3.97}$$

Cette dernière formule est la troisième équation consistante qui est intriquée avec les deux autres équations. L'introduction de la mesure de Bregman, permet d'obtenir les solutions du problème pour plusieurs mesures de pertes classiques qui sont des divergences de Bregman. Le Tableau 3.1 résume quelques divergences connues. Ces trois équations consistantes Eq. 3.92, Eq. 3.94 et Eq. 3.97 garantissent que la borne débit-distorsion est atteinte. Supposons, que l'on veuille s'approcher de cette borne avec une fonction d'assignement $p(\hat{x} \mid x)$ déterministe. Cela revient à choisir pour chaque réalisation x, le mot de code ou la reconstruction $g \circ f(x)$ le plus probable :

$$g \circ f(x) = \max_{\hat{x}} p(\hat{x}) e^{-\beta d(x, \hat{x})} \tag{3.98}$$

Ainsi, la probabilité conditionnelle $p(\hat{X} \mid X)$ suit la définition donnée dans l'Eq. 3.87. Cet assignement déterministe permet de construire des groupes où chacun d'entre eux est attaché à un $\hat{x}$. Dénotons ces groupes par $C(\hat{x}) = \{x \in \mathcal{X}, g \circ f(x) = \hat{x}\}$. Alors, la seconde équation consistante Eq. 3.94, se réécrit :

$$p(\hat{x}) = \sum_{x \in C(\hat{x})} p(x) \tag{3.99}$$

Enfin, la troisième équation Eq. 3.97, correspond à calculer la moyenne de chaque groupe pour calculer leur centre de gravité.

$$\hat{x} = \frac{1}{p(\hat{x})} \sum_{x \in C(\hat{x})} p(x)x \tag{3.100}$$

Ces trois équations consistantes permettent de déterminer la fonction de codage-décodage qui se rapproche au plus de la courbe débit-distorsion. Nous notons, que dans certains cas, l'espace des observations n'est pas un espace vectoriel et la mesure de pertes n'est plus une divergence de Bregman. Dans, ce cas la moyenne est approximée par une médiane (Kaufman & Rousseeuw, 1990), en espérant que cette approximation minimise la fonction de pertes d. Alors, le mot de code $\hat{x}$ est choisi dans l'ensemble $C(\hat{x})$ tel que :

$$\hat{x} = \arg \min_{y \in C(\hat{x})} \sum_{x \in C(\hat{x})} d(x, y)p(x) \tag{3.101}$$

Cette dernière procédure, cherche à calculer l'objet médian qui est le plus proche en moyenne de tous les autres. Comme nous l'avons vu, ces équations consistantes sont intriquées et ne peuvent être inférées en même temps. Nous présentons dans le chapitre suivant, une méthode qui permet de résoudre ces équations de manière itérative.

3.4.2.2 Algorithme d'Espérance-Maximisation

Dempster et al. (1977) introduisent l'algorithme d'Espérance-Maximisation pour le calcul d'un mélange de lois statistiques. Notre lagrangien $\mathcal{L}$ dépend uniquement de $p(\hat{X} \mid X)$ et de $\mathcal{X}$. Le but final de la minimisation du lagrangien est de trouver la probabilité optimale qui donnera un point de la courbe débit-distorsion en fonction du paramètre de Lagrange β.
L'algorithme Espérance-Maximisation consiste à fixer un des deux paramètres tour à tour dans le but de calculer l'autre en s'aidant des équations consistantes. Banerjee et al. (2004a) utilise cette méthode pour calculer les inconnus du problème de minimisation. La première étape de l'algorithme est l'Espérance et consiste à fixer la probabilité d'assignement $p(\hat{X} \mid X)$ et $p(\hat{X})$. Ainsi, l'ensemble des mots de code $\hat{\mathcal{X}}$ est obtenu par l'Eq. 3.97. La seconde étape est la Maximisation et elle consiste à fixer $\hat{\mathcal{X}}$, pour obtenir successivement $p(\hat{X} \mid X)$ et $p(\hat{X})$ via les équations Eq. 3.92 et Eq. 3.94. Dans notre cas, ce n'est pas tout à fait une maximisation, mais dans le cas d'un assignement déterministe cette étape le devient. Alors, en itérant ces deux étapes successivement, Banerjee et al. (2004a) prouvent que l'algorithme converge vers un minimum local de $\mathcal{L}$. L'algorithme est décrit plus précisément dans Algorithm 1. Dans cet algorithme, nous voyons que nous pouvons restreindre la taille de l'espace des représentants $\hat{\mathcal{X}}$ à k éléments. Il faut régler conjointement k et β pour obtenir la sortie souhaitée. Il est souhaitable de prendre $k = n$, pour

Entrées : l'ensemble $\mathcal{X} = \{x_i\}_{i=1}^n$, la divergence de Bregman D_ϕ, le nombre de groupes k et la distribution $\{p(x_i)\}_{i=1}^n$
Sorties : l'ensemble $\hat{\mathcal{X}} = \{\hat{x}_i\}_{i=1}^k$, les probabilités $\{\{p(\hat{x}_i \mid x_j)\}_{i=1}^k\}_{j=1}^n$ et $\{p(\hat{x}_i)\}_{i=1}^k$
Initialisation : choisir les sorties quelconques telles qu'elles respectent l'intégration à 1.
while convergence **do**
  étape de Maximisation
  for $j = 1$ to n **do**
    for $i = 1$ to k **do**
      $p(\hat{x}_i \mid x_j) = \frac{p(\hat{x}_i)e^{-\beta d(x_j, \hat{x}_i)}}{\sum_{\hat{x}} p(\hat{x})e^{-\beta d(x_j, \hat{x})}}$
    end for
  end for
  for $i = 1$ to k **do**
    $p(\hat{x}_i) = \sum_x p(\hat{x}_i \mid x)p(x)$
  end for
  étape d'Espérance
  for $i = 1$ to k **do**
    $\hat{x}_i = \sum_x p(x \mid \hat{x}_i)x$
  end for
end while

Algorithm 1 – Algorithme Espérance-Maximisation pour le calcul de la courbe débit-distorsion

calculer la courbe débit-distorsion, puisque dans le cas où $k < n$ il n'est pas possible d'obtenir un codage sans perte. L'autre paramètre β règle le compromis qu'il y a entre le débit et les pertes. Quand $\beta \to 0$, les pertes deviennent grandes et le débit petit. Au contraire, quand $\beta \to \infty$, on se rapproche d'un codage sans perte. Une fois que l'algorithme a convergé, les sorties $p^*(\hat{X}^*, X), p^*(\hat{X}^*), \hat{X}^*$ sont accessibles et il est possible de calculer un point de la fonction débit-distorsion associé au paramètre de Lagrange β.

$$R(\beta) = \sum_{x,\hat{x}^*} p^*(\hat{x}^* \mid x)p(x) \log \frac{p^*(\hat{x}^* \mid x)}{p^*(\hat{x}^*)} \tag{3.102}$$

$$D(\beta) = \sum_{x,\hat{x}^*} p^*(\hat{x}^* \mid x)p(x)D_\phi(x, \hat{x}^*) \tag{3.103}$$

Si nous prenons $k < n$ et $\beta >> 1$, nous retombons sur l'algorithme des k-moyennes floues (Bezdek, 1981). D'autre part, nous avons vu au chapitre précédent que nous pouvons nous restreindre aux fonctions d'assignement déterministe. Dans ce cas, en prenant $\beta >> 1$ et $k < n$, l'algorithme présenté devient équivalent à l'algorithme des k-moyennes (MacQueen, 1965) et à l'algorithme de Lloyd-Max (Gray & Neuhoff, 1998) qui sont des algorithmes de groupage et de quantification.
Enfin, ce type d'algorithme converge vers un minimum local. Pour résoudre ce problème, des méthodes par recuit simulé (Rose, 1998) peuvent être employées pour garantir la convergence vers un minimum global. Nous présenterons cette méthodes ultérieurement.

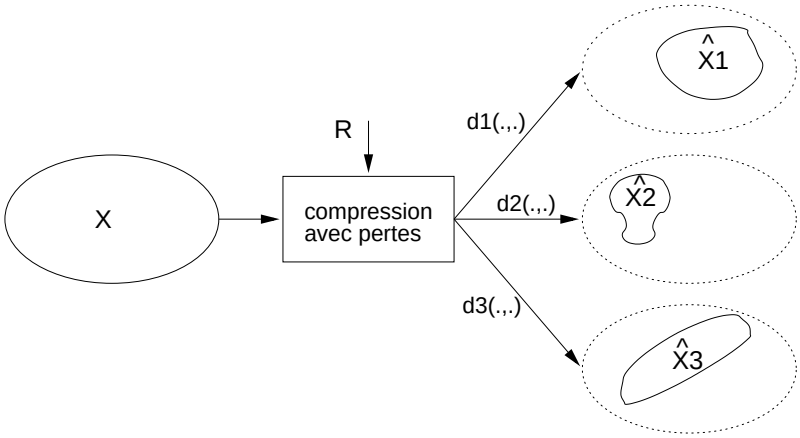


FIG. 3.6 – Ce schéma représente schématiquement l'information contenue dans la variable X. D'autre part, il est clair que l'information extraite dans $\hat{X}_i$ dépend de la distorsion $d_i(.,.)$ choisie. A un même débit, plusieurs types d'information peuvent être extraits.

3.4.3 Extraction d'information et compression avec pertes

En fin de compte, l'algorithme décrit précédemment permet de clustériser et de faire de la quantification vectorielle qui vont dans le même sens (cf. §2.6.3). Comme, nous l'avons noté au §3.4.1, quand la fonction d'assignement est déterministe, l'information de $\hat{X}$ est une sous-partie de l'information véhiculée par X. Donc, la théorie débit-distorsion est un cadre pour l'extraction d'information. En effet, cette théorie nous permet de mesurer la quantité d'information extraite et de la qualifier via la fonction de pertes. Enfin, vis à vis de la conceptualisation du problème général (cf. §2.7), nous comprenons mieux pourquoi les algorithmes de groupage sont utilisées pour l'indexation de bases de données. En effet, la création d'un index peut être vue comme une compression avec pertes. Cependant, la fonction de pertes ou la distance utilisée détermine le type d'information qui sera extrait. De plus, il est difficile de prévoir a priori la quantité d'information nécessaire à la discrimination des objets pour différentes requêtes. Cette remarque est schématisée dans la Figure 3.6. En fonction, du débit et de la distance différentes parties de l'information sont extraites.

3.5 Complexité de Kolmogorov comme mesure de l'information

Dans ce chapitre nous décrivons la complexité de Kolmogorov comme mesure de l'information. A partir de cette définition, nous décrirons comment les principes énoncés dans la théorie du codage de Shannon se retrouvent dans ce cadre théorique. Nous nous appuierons sur ces mesures de l'information pour présenter des moyens de modéliser et d'extraire l'information. Enfin, nous finirons par décrire une mesure de similarité exclusivement fondée sur les quantités d'information.

3.5.1 Complexité de Kolmogorov

La complexité de Kolmogorov $K(x)$ d'un objet x se définit comme la longueur de la plus petite description de x. Cette mesure de l'information est vue comme la longueur du plus petit programme qui est capable de générer l'objet x. Ce programme peut être écrit

en c/c++, Java ou n'importe quel langage et les longueurs des programmes résultants ne différent que d'une constante indépendante de l'objet traité. Kolmogorov (1965) formalise cette mesure en considérant une machine de Turing universelle U qui agit sur n'importe quelle chaîne ou programme $p \in \{0,1\}^*$ de longueur finie et s'arrête. De plus, il considère que le programme p appartient à un ensemble de codes préfixes P. L'ensemble est préfixe si chaque élément est décodable d'une seule manière, c'est-à-dire qu'aucun programme de l'ensemble n'est le préfixe d'un autre élément de l'ensemble. Si l'ensemble est préfixe, alors les longueurs de code associées à chaque élément de P vérifient l'inégalité de Kraft (Eq :3.66). La complexité de l'objet en rapport à cette machine U est donnée par :

$$K(x) = \min_{p \in P} \{l(p) : U(p) = x, \} \tag{3.104}$$

où $l(.)$ est la longueur d'un programme. La machine de Turing exécute le programme p qui donne en sortie l'objet x. Grünwald & Vitanyi (2004) note ce plus petit programme x^* tel que :

$$x^* = \arg\min_{p \in P} \{l(p) : U(p) = x\} \tag{3.105}$$

$$K(x) = l(x^*) \tag{3.106}$$

Cette définition est dépendante de la machine de Turing employée. Considérons, deux machine U_1 et U_2, alors pour n'importe quel objet x, Kolmogorov (1965) prouve que $\mid K_{U_1}(x) - K_{U_2}(x) \mid \leq C$, où C est une constante indépendante de l'objet x. Cela vient du fait qu'une machine de Turing préfixe peut émuler n'importe quelle autre machine préfixe. Alors, la complexité de Kolmogorov est toujours définie à une constante et dans ce cas nous utiliserons la notation :

$$K_{U_1}(x) \stackrel{+}{=} K_{U_2}(x) \tag{3.107}$$

Cette complexité peut se définir également sur un ensemble fini $S \subset \{0,1\}^*$ ou sur une fonction calculable f. Ainsi $K(S)$ est la longueur du plus court programme qui permet de générer l'ensemble S et $K(f)$ est la longueur du plus court programme qui permet de calculer la fonction f. Il est à noter que la complexité de Kolmogorov n'est pas une mesure calculable. En effet, il n'existe aucun programme qui en donnant un objet en entrée ressort la complexité de Kolmogorov et stoppe.
Par construction, la complexité de Kolmogorov vérifie l'inégalité de Kraft (Eq. 3.66), puisqu'elle est construite sur un ensemble vérifiant la propriété d'un code préfixe. Si cette inégalité est atteinte, alors l'ensemble P est dit complet, c'est-à-dire qu'il est impossible de rajouter un programme à celui-ci sans que l'ensemble perde la propriété d'être préfixe. Cette égalité permet de définir une distribution universelle (Li & Vitanyi, 1997) sur les objets de $\{0,1\}^*$ en suivant l'égalité $p(x) = 2^{-K(x)}$. Cette distribution universelle signifie que les objets les moins complexes ont plus de probabilités d'apparaître dans l'univers.

3.5.2 Moyenne des complexités et entropie de Shannon

Grünwald & Vitanyi (2004) démontrent les liens qui existent entre la moyenne des complexités de Kolmogorov et l'entropie de Shannon. Ils déduisent ainsi deux bornes qui permettent de lier ces deux mesures de l'information. Ils considèrent une source aléatoire X distribuée selon la probabilité $p(X)$. On peut se poser la question de savoir si le code engendrée par K et indépendant de la probabilité p et s'il est un code

universel. D'une part, comme la complexité de Kolmogorov vérifie l'inégalité de Kraft, la complexité moyenne est supérieure à l'entropie d'après le premier théorème de codage de Shannon. D'autre part, Grünwald & Vitanyi (2004) démontrent que la complexité moyenne est inférieure à l'entropie de Shannon à une constante prés qui dépend de p. Ces deux inégalités sont résumées par :

$$H(X) \leq \sum_x p(x)K(x) \stackrel{+}{\leq} H(X) + K(p) \tag{3.108}$$

où $K(p)$ est la complexité de la distribution de probabilité. Cette inégalité montre que pour des distributions simples la complexité moyenne est proche de l'entropie de Shannon. Cependant dans le cas contraire, ces deux quantités peuvent être éloignées. Prenons, le cas où la distribution est déterministe $p(x) = 1$, alors l'entropie $H(X) = 0$ et la complexité moyenne est $K(x)$. La différence devient grande et nous remarquons que $K(p) \stackrel{+}{\geq} K(x)$. Cet inégalité se justifie car dans ce cas précis la connaissance de p permet de trouver x.
Comme Shannon a étendu l'entropie à l'information mutuelle, Zvonkin & Levin (1970) ont étendu la complexité à l'information mutuelle algorithmique en définissant la complexité conditionnelle de Kolmogorov définie entre deux objets x et y par :

$$K(x \mid y) = \min_{p \in \{0,1\}^*} \{l(p) : U(< p, y >) = x\} \tag{3.109}$$

Cette définition est fondée sur l'existence d'un programme p qui prend en entrée l'objet y et ressort x et stoppe. Dans le cas où l'objet en entrée est une chaîne vide, nous retombons sur la complexité de Kolmogorov. Zvonkin & Levin (1970) démontrent la propriété d'additivité des complexités exprimée par :

$$K(x, y) \stackrel{+}{=} K(x) + K(y \mid x^*) \stackrel{+}{=} K(y) + K(x \mid y^*) \tag{3.110}$$

Par cette propriété, l'information mutuelle algorithmique est définie par :

$$I(x : y) = K(x) - K(x \mid y^*) \stackrel{+}{=} K(x) + K(y) - K(x, y) \stackrel{+}{=} K(y) - K(y \mid x^*) \tag{3.111}$$

Cette mesure est symétrique et représente l'information algorithmique mutuelle partagée par les deux objets, par analogie à l'information mutuelle (Eq. 3.69). Grünwald & Vitanyi (2004) utilisent cette définition pour lier la moyenne des informations mutuelles algorithmiques à l'information mutuelle. Ils considèrent une distribution conjointe $p(X, Y)$ entre deux variables et montrent que :

$$I(X, Y) - K(p) \stackrel{+}{\leq} \sum_x \sum_y p(x, y) I(x : y) \stackrel{+}{\leq} I(X, Y) + 2K(p) \tag{3.112}$$

Comme pour la complexité moyenne, quand la probabilité conjointe est peu complexe ces deux mesures de l'information mutuelle sont équivalentes. En résumé, les mesures de l'information par entropie et par complexité sont équivalentes sous des conditions de faible complexité des distributions.

3.5.3 Le principe LDM dans la théorie de Kolmogorov

Dans un cas l'entropie de Shannon est une approximation quand la distribution n'est pas connue et d'autre part la complexité de Kolmogorov n'est pas une fonction calculable. Ainsi, ces deux mesures ne peuvent être utilisée directement pour la construction

d'une méthode de codage universelle. Cependant, l'introduction des codes universels (cf. §3.3.4) peut être vue comme une généralisation de la mesure de Shannon et comme une approximation de la complexité de Kolmogorov. Ce dernier introduit le codage universel en disant :

> Une méthode de codage universel qui permet de transmettre n'importe quel message de taille n dans un alphabet de s lettres avec pas plus de nh bits (h est l'entropie empirique), n'est pas forcément excessivement complexe ; en particulier, il n'est pas nécessaire de commencer par déterminer les fréquences d'apparition pour le message entier.

Pour formaliser cette constatation, il suppose une énumération de codages préfixes $D_1, ...$ de fonctions de longueurs $L_1, ...$ respectives. Il introduit la longueur de code nécessaire à coder x en supposant chaque codage :

$$L_i(x) = \min_y \{l(y) : D_i(y) = x\} \tag{3.113}$$

Ainsi, x peut être encodé en deux parties. En premier, un entier est encodé pour déterminer le codage, et ensuite x sachant le codage est encodé. Cela revient à un codage en deux parties de longueur totale déterminée par la plus petite somme :

$$L(x) = \min_k \{L(k) + L_k(x)\} \tag{3.114}$$

où $L(k)$ est la longueur nécessaire à encoder l'entier k. D'autre part, une code de longueur $\tilde{L}(x)$ est universel par rapport à la classe de code $D_1, D_2, ...$ si celui-ci compresse aussi bien n'importe quelle séquence que les meilleurs codages D_k associés à chacune des réalisations. Cette définition se traduit sur des signaux dont la taille n tend vers l'infini par :

$$\lim_{n\to\infty} \tilde{L}(x) \geq L_k(x) \tag{3.115}$$

De nombreux codes vérifient cette propriété et en particulier les codes en deux parties. Par la suite, cette définition de code en deux parties a été étendue aux modèles qui permettent de générer les objets. Soit un modèle $\mathcal{M}$ qui permet de générer x par la connaissance de y tel que $\mathcal{M}(y) = x$. Alors la meilleure description en deux parties de x a pour longueur :

$$L(x) = \min_{\mathcal{M}} \{K(y) + K(\mathcal{M}) : \mathcal{M}(y) = x\} = \min_{\mathcal{M}} \{K(x \mid \mathcal{M}) + K(\mathcal{M})\} \tag{3.116}$$

En confrontant cette définition à celle du LDM, nous notons que ce principe est équivalent. Dans le cas de modèles paramétriques, $K(\mathcal{M})$ s'apparente à la redondance de codage exprimée dans Eq. 3.80. Et le premier terme $K(x \mid \mathcal{M})$ s'apparente à la vraisemblance du modèle exprimée par $-\log p(x \mid M)$. Ainsi, nous voulons mettre en avant que tous les principes d'estimation peuvent se réinterpréter en termes de complexité minimale.
Cette longueur de code en deux parties est universelle mais reste supérieure à $K(x)$ puisque :

$$K(x) \leq K(x, \mathcal{M}) \buildrel + \over = K(x \mid \mathcal{M}) + K(\mathcal{M}) \tag{3.117}$$

En fin de compte, le principe d'une description en deux parties à longueur minimale (Eq. 3.116) vise à s'approcher de l'égalité $K(x) \buildrel + \over = K(x \mid \mathcal{M}) + K(\mathcal{M})$.

3.5.4 La fonction de structure de Kolmogorov

La fonction de structure de Kolmogorov a été introduite pour le codage avec une perte d'information. Elle s'apparente à un estimateur par Maximum de Vraisemblance et cette fonction est définie par :

$$h_x(R) = \min_S \{\log \mid S \mid : x \in S, K(S) \leq R\} \tag{3.118}$$

Ici, S est un ensemble qui contient x. Le but de cette fonction est d'évaluer le plus petit ensemble qui contient x. Dans certains cas, cet ensemble peut être réécrit pour des modèles probabilistes sous la forme :

$$h_x(R) = \min_{\mathcal{M}} \{-\log p(x \mid \mathcal{M}) : K(\mathcal{M}) \leq R\} \tag{3.119}$$

Cette fonction peut être associée à la fonction débit-distorsion présentée dans l'Eq. 3.86. Pour cela, Vereshchagin & Vitanyi (2004) introduisent une distorsion associée à un ensemble de modèles probabilistes comme étant :

$$d(x, \mathcal{M}) = -\log p(x \mid \mathcal{M}) \tag{3.120}$$

Les modèles qui assument cette définition de la distorsion font partie de la classe des modèles de Kolmogorov admettant des pertes. Cette distorsion représente l'indétermination sur x sachant le modèle. Par exemple, prenons l'ensemble des x tels que $d(x, \mathcal{M}) = r$. Connaissant ce modèle chaque élément peut être codé sur r bits ce qui correspond à avoir un ensemble de 2^r éléments. Une procédure de codage avec pertes de x pourrait donc consister à coder $\mathcal{M}$ avec R bits puisque $K(\mathcal{M}) \leq R$. Ensuite sachant que la distorsion est r, on peut reconstruire l'ensemble précédent. Enfin, en prenant un élément au hasard dans cet ensemble on a une version approchée de x.
Avec cette définition de la distorsion, une source X suivant une distribution $p(X)$ admet une courbe débit-distorsion $D_n^*(R)$ qui dépend du nombre de réalisations n (cf. §3.4.1). Soit, x^n une suite de n réalisations $\{x_1, x_2, ..., x_n\}$ i.i.d. Grünwald & Vitanyi (2004) démontrent les inégalités suivantes :

$$\frac{1}{n}\sum_{x^n} p(x^n) h_{x^n}(nR + K(p, d, n, R)) \leq D_n^*(R) \leq \frac{1}{n}\sum_{x^n} p(x^n) h_{x^n}(nR) \tag{3.121}$$

La moyenne de la fonction de structure s'approche de la fonction débit-distorsion. Nous notons que la fonction de structure est calculée sur les suites de réalisations ainsi elle permet de prendre en compte des propriétés de dépendances entre réalisations. Quand n tend vers l'infini, la fonction $D_n^*(R)$ tend vers la limite $D^*(R)$ qui suit l'égalité suivante :

$$\lim_{n\to\infty} \frac{1}{n}\sum_{x^n} p(x^n) h_{x^n}(nR) = D^*(R) \tag{3.122}$$

Cette définition est en fait valable pour n'importe quel type de distance, puisque que l'on peut déduire des modèles probabilistes par $p(x \mid \mathcal{M}) \propto e^{-d(x,\mathcal{M})}$.
Comme nous l'avons décrit pour la fonction débit-distorsion dans l'Eq. 3.89, cette fonction de structure peut se réécrire sous forme laplacienne, avec β un paramètre de compromis. Ainsi la minimisation devient :

$$\min_{\mathcal{M}} \{-\log p(x^n \mid \mathcal{M}) + \beta K(\mathcal{M})\} \tag{3.123}$$

Si ce minimum est atteint en $\hat{\mathcal{M}}$ alors on obtient la fonction de structure paramétrée en fonction de β et qui est définie telle que :

$$R^{\beta} = K(\hat{\mathcal{M}}) \tag{3.124}$$

$$h^{\beta}_{x^n} = -\log p(x^n \mid \hat{\mathcal{M}}) \tag{3.125}$$

Nous notons que dans le cas où $\beta = 1$, nous retombons sur le principe LDM de codage en deux parties. Ainsi, en prenant un codage universel qui satisfait la minimisation de l'Eq. 3.123, nous sommes sûr que nous nous approchons du point de la courbe débit-distorsion obtenue avec une compromis équitable entre la distorsion et le débit. Cette vision du codage en deux parties introduit un nouveau principe pour le codage avec pertes. Concrètement, par ce principe on obtient une moyen de séparer l'information en deux parties distinctes telles que la combinaison de ces deux quantités permettent la reconstruction de $(x, \mathcal{M})$ avec $K(x, \mathcal{M})$ bits d'information.
En conclusion de ce paragraphe, le codage en deux parties lié aux codes universels est aussi une procédure de compression avec pertes. D'une part la partie liée au modèle est vue comme l'information pertinente qu'il faut conserver, tandis que l'autre partie d'information est le résultat de la distorsion. En effet, cette seconde partie est considérée comme uniquement aléatoire puisqu'elle n'est pas nécessaire pour l'approximation du signal d'origine. Du fait que le principe LDM tire ses fondements de la théorie de codage de Kolmogorov, ce principe permet d'extraire une quantité d'information suffisante du signal d'origine.

3.5.5 Mesure de similarité fondée sur l'information

Nous avons vu au §2.5.3, que plusieurs similarités sont couramment utilisées pour comparer deux objets. En particulier, nous avons vu les distances fondées sur les longueurs des codes. Nous nous intéressons ici à décrire les propriétés de la métrique de similarité (Li, 2003). Tout d'abord, ils définissent la distance informationnelle $\mathcal{E}(x, y)$ entre deux objets x et y comme le plus petit programme capable de calculer x à partir de y et y à partir de x. Ainsi, ce plus petit programme prendra avantage des redondances qui existent entre les deux objets. Bennett et al. (1998) montrent que la longueur de ce programme s'exprime par :

$$\mathcal{E}(x, y) \stackrel{+}{=} \max\{K(x \mid y), K(y \mid x)\} \tag{3.126}$$

Cette métrique est une distance puisque Li et al. (2004) montrent qu'elle vérifie les propriétés énoncées au §2.5.3.
Considérons l'ensemble des distances admissibles, qui est défini comme l'ensemble des distances D telles que pour tout x, $D(x, .)$ soit une fonction de longueur de code qui vérifie l'inégalité de Kraft. En d'autres termes, cette fonction de distance $D(x, .)$ correspond à un code préfixe. Alors, la distance informationnelle $\mathcal{E}$ minimise n'importe quelle distance appartenant à l'ensemble des distances admissibles, ce qui se traduit par :

$$\mathcal{E}(x, y) \stackrel{+}{\leq} D(x, y) \tag{3.127}$$

Par cette inégalité, nous savons que si deux objets sont proches avec la mesure D alors elles seront proches avec $\mathcal{E}$. Pour mesurer la similarité entre deux objets, différents types d'information peuvent être considérés. Si par un type d'information les objets sont similaires, alors la distance $\mathcal{E}$ ne contredira pas ce fait. C'est comme si elle regardait sur quels

types d'information les objets sont similaires. Néanmoins cette mesure reste relative. Par exemple deux chaînes de longueurs 10^6 partageant un distance 1000 sont sûrement plus similaires que deux chaînes de longueurs 1000 qui partagent cette même distance. Ainsi, Li et al. (2004) proposent de normaliser cette distance en considérant l'ensemble des distances normalisées qui prennent leurs valeurs dans l'intervalle $[0, 1]$. Une telle distance d vérifie de plus l'inégalité de Kraft suivante :

$$\sum_y 2^{-d(x,y)K(x)} \stackrel{<}{=} 1 \tag{3.128}$$

A partir de cette définition, une distance convient et s'exprime par :

$$d(x,y) \stackrel{+}{=} \frac{\mathcal{E}(x,y)}{\max\{K(x), K(y)\}} = \frac{\max\{K(x \mid y), K(y \mid x)\}}{\max\{K(x), K(y)\}} \tag{3.129}$$

Cette distance de similarité minimise l'ensemble des distances normalisées comme $\mathcal{E}$. Ainsi, cette mesure de similarité est aussi universelle sur l'ensemble des distances normalisées calculables. Du fait de cette propriété d'universalité nous serons amenés à utiliser cette distance.

3.6 Résumé

Dans ce chapitre, nous avons souligné que les techniques de modélisation probabiliste et de codage sont équivalentes. Ainsi nous utiliserons indifféremment ces méthodologies de modélisation. D'autre part, nous avons mis en avant que les principes de longueur de code minimum permettent de déduire des modèles efficaces pour la représentation des objets et l'extraction d'information. Ce cadre théorique permet de formaliser, la fouille d'information et l'indexation de bases de données comme un canal de communication entre les données et un utilisateur. Ce formalisme, permet d'identifier les quantités d'information extraites et servant aux recherches par le contenu. En résumé, les points suivants ont été abordés :

- les modèles stochastique et les champs de Gibbs-Markov pour la modélisation de signaux multidimensionnels. Les champs aléatoires serviront pour la modélisation des STIS aux §4.3.1 et 5.3.2 ;
- l'estimation de paramètres et la sélection de modèle ;
- l'extraction d'information par le codage avec pertes, tel que la quantification vectorielle ou le clustering. Cette approche sera la base du §4 ;
- la modélisation par le codage sans perte de Shannon et Kolmogorov et les équivalences avec l'estimation bayésienne ;
- le codage en deux parties vu comme une compression avec pertes. Cette dernière constatation sera largement exploitée au §5.

Deuxième partie

Méthodes entropiques et basées sur les complexités pour l'extraction d'information des STIS

Chapitre 4

Extraction d'information des STIS par compression avec pertes

Dans ce chapitre, nous décrivons une méthode pour l'extraction automatique d'information des STIS fondée sur la compression avec pertes décrite au §3.4.3. Le but est d'extraire le minimum d'information nécessaire à décrire au mieux les données. Nous présentons dans un premier temps le principe d'*Information Bottleneck* (IB) et les algorithmes associés. Nous verrons par la suite que ce principe permet d'extraire l'information d'intérêt portée par une variable aléatoire. La quantité d'information extraite est optimale au sens d'un critère heuristique créé dans le cadre de ce travail. Puis, nous appliquons cette méthodologie pour extraire la quantité d'information optimale liée aux structures spatio-temporelles, et nous utilisons des modèles de champs aléatoires.
D'autre part, nous introduisons le principe de *Multi-Information Bottleneck* (MIB) déduit du principe précédent. Ce principe permet de prendre en compte plusieurs types d'information. Nous décrivons les algorithmes associés et nous les appliquons aux STIS pour extraire l'information de couleur et d'évolution de texture. Cette méthodologie généralise la première méthode présentée.
Finalement, nous donnons un principe théorique qui permet d'aborder l'extraction d'information sous un nouveau point de vue en considérant la variation de l'information. Nous verrons que ce principe est le dual du principe IB.

4.1 Le principe d'*Information Bottleneck*

Dans cette partie, nous décrivons le principe d'*Information Bottleneck* pour l'extraction d'information. Ce principe rentre dans le cadre de la théorie débit-distorsion présentée au §3.4.1. Nous décrivons dans un premier temps ce principe, avant de présenter les algorithmes qui en découlent.

4.1.1 Description du principe d'*Information Bottleneck*

Tishby et al. (1999) introduisent le principe d'*Information Bottleneck* comme un problème de communication avec pertes. La problématique liée à ce principe est de produire une quantification d'une variable X sous la contrainte d'une distorsion particulière. La contrainte est que la quantification doit conserver le plus possible d'information contenue dans une troisième variable aléatoire Y. En termes de communication, nous souhaitons

faire passer le plus d'information possible que X possède par rapport à Y au travers d'un goulot d'étranglement formé par la quantification. En termes d'extraction d'information, si Y représente une information a priori importante, alors ce principe permet d'extraire de X l'information d'importance. Nous notons $\tilde{X}$ la représentation compacte de l'information d'importance contenue dans X. La mesure des pertes entre X et $\tilde{X}$ correspond alors à la perte d'information d'importance occasionnée par la quantification et s'exprime par :

$$D(X, \tilde{X}) = I(X, Y) - I(\tilde{X}, Y) \tag{4.1}$$

Dans le cadre de la théorie débit-distorsion, le principe s'exprime alors comme :

$$\min_{p(\tilde{x}|x), \tilde{\mathcal{X}}} I(X, \tilde{X}) \quad \text{s.c.} \quad D(X, \tilde{X}) \leq D \tag{4.2}$$

où $\tilde{\mathcal{X}}$ est l'espace de reproduction. Plus clairement, ce principe veut transmettre le minimum d'information contenue dans X tout en conservant le maximum d'information contenue dans Y. Cette minimisation admet une représentation lagrangienne comme l'Eq. 3.89 et s'exprime par :

$$\min_{p(\tilde{x}|x), \tilde{\mathcal{X}}} I(\tilde{X}, X) + \beta \Big\{ I(X, Y) - I(\tilde{X}, Y) \Big\} \tag{4.3}$$

Dans cette minimisation, l'information mutuelle $I(X, Y)$ est constante car elle ne dépend pas des arguments de minimisation. Cela implique que cette quantité d'information est connue a priori. Par conséquent, nous pouvons supprimer ce terme de minimisation et réécrire le principe IB dans sa formulation courante :

$$\min_{p(\tilde{x}|x), \tilde{\mathcal{X}}} I(\tilde{X}, X) - \beta I(\tilde{X}, Y) \tag{4.4}$$

où $p(X)$ et $p(Y \mid X)$ sont connues a priori. Nous donnons une représentation abstraite de la répartition de l'information en fonction du paramètre β dans la Figure 4.1. Quand β tend vers 0 la quantité d'information portée par $\tilde{X}$ est nulle. Au contraire, quand β est grand, l'information portée par $\tilde{X}$ tend à recouvrir celle de X. On se rapproche d'un compression sans perte.
Nous avons vu que cette formulation est équivalente à la définition d'une courbe débit-distorsion. Pour pouvoir appliquer les méthodes de quantification présentées au §3.4.2, nous montrons que la distorsion est une divergence de Bregman.

4.1.2 Equations consistantes

Banerjee et al. (2004b) démontrent que la mesure de pertes de l'Eq. 4.1 peut s'exprimer sous la forme d'une divergence de Bregman. Alors le principe d'*Information Bottleneck* peut être traité comme une minimisation d'une fonction débit-distorsion générale. Pour commencer, nous faisons l'hypothèse que Y et $\tilde{X}$ sont indépendants connaissant X. Cette supposition s'explique par le fait que connaissant X, l'information de Y ne dépend plus d'une sous partie de l'information de X contenue dans $\tilde{X}$, puisque celle-ci est déjà connue. Cette supposition se traduit par la relation markovienne $Y \leftrightarrow X \leftrightarrow \tilde{X}$ et se traduit statistiquement par :

$$p(y, \tilde{x} \mid x) = p(y \mid x) p(\tilde{x} \mid x) \tag{4.5}$$

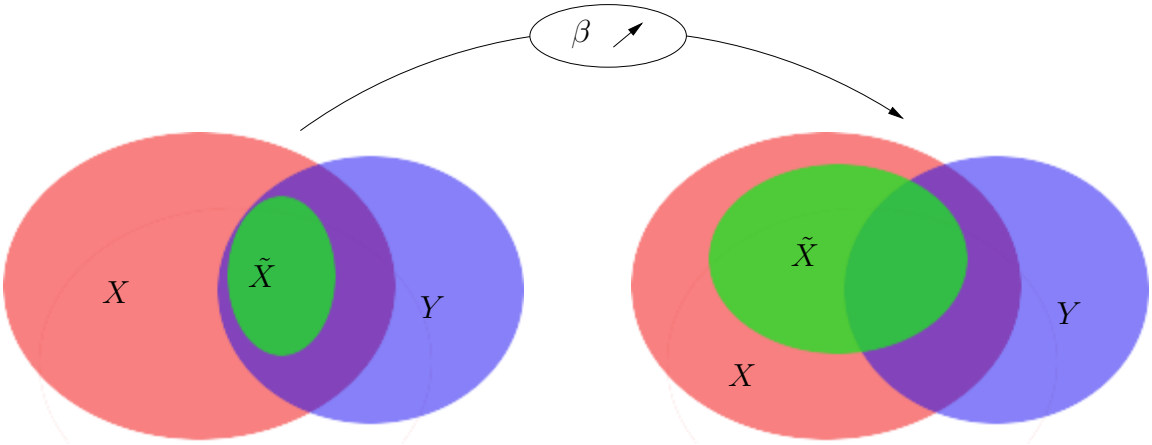


FIG. 4.1 – Nous donnons une représentation abstraite de l'information sous forme d'ensembles. L'information mutuelle $I(X,Y)$ correspond à l'intersection des deux ensembles X et Y. Nous présentons l'évolution de l'information $\tilde{X}$ quand β augmente. Quand $\beta \to \infty$, l'ensemble $\tilde{X}$ a tendance à recouvrir X, ce qui donne une distorsion s'approchant de 0.

A partir de cette hypothèse réaliste, nous donnons une formulation équivalente de la fonction de pertes :

$$
\begin{aligned}
D(X,\tilde{X}) &= I(X,Y) - I(\tilde{X},Y) && (4.6)\\
&= \sum_{y,x} p(y,x) \log \frac{p(y \mid x)}{p(y)} - \sum_{y,\tilde{x}} p(y,\tilde{x}) \log \frac{p(y \mid \tilde{x})}{p(y)} \\
&= \sum_{y,\tilde{x},x} p(x,\tilde{x},y) \log \frac{p(y \mid x)}{p(y \mid \tilde{x})} \\
&= \sum_{y,\tilde{x},x} p(\tilde{x},y \mid x) p(x) \log \frac{p(y \mid x)}{p(y \mid \tilde{x})} \\
&= \sum_{y,\tilde{x},x} p(\tilde{x} \mid x) p(y \mid x) p(x) \log \frac{p(y \mid x)}{p(y \mid \tilde{x})} \\
&= \sum_{\tilde{x},x} p(\tilde{x},x) \sum_{y} p(y \mid x) \log \frac{p(y \mid x)}{p(y \mid \tilde{x})} \\
&= \mathbb{E}_{X,\tilde{X}} \Big[d_{KL}(p(Y \mid X) \mid p(Y \mid \tilde{X})) \Big] && (4.7)
\end{aligned}
$$

Nous avons vu que la divergence de Kullback-Leibler est une divergence de Bregman pour les espaces d-Simplex dans le Tableau 3.1. Pour satisfaire ces conditions, Banerjee et al. (2004a) introduisent une variable Z (respectivement $\tilde{Z}$) prenant ses valeurs dans l'espace des probabilités conditionnelles $\mathcal{Z} = \{p(Y \mid x)\}_{x \in \mathcal{X}}$ (respectivement $\tilde{\mathcal{Z}} = \{p(Y \mid \tilde{x})\}_{\tilde{x} \in \tilde{\mathcal{X}}}$) qui est un espace d-Simplex. Comme Z et $\tilde{Z}$ sont des statistiques suffisantes, il devient évident de considérer que $p(X,\tilde{X}) = p(Z,\tilde{Z})$, $p(X) = p(Z)$ et $p(\tilde{X}) = p(\tilde{Z})$. Alors, le principe IB (Eq. 4.3) se réécrit comme une problème débit-distorsion classique en prenant comme mesure de pertes la divergence de Kullback-Leibler :

$$
\min_{p(\tilde{z}|z),\tilde{\mathcal{Z}}} I(\tilde{Z},Z) + \beta \mathbb{E}_{Z,\tilde{Z}} \Big[d_{KL}(Z \mid \tilde{Z}) \Big] \qquad (4.8)
$$

Par conséquent, nous pouvons calculer les trois équations consistantes décrites au §3.4.2.1. Nous réécrivons les équations Eq. 3.92, Eq. 3.94 et Eq. 3.97 :

$$\begin{aligned} p(\tilde{z} \mid z) &= \frac{p(\tilde{z})}{N(z,\beta)} e^{-\beta d_{KL}(z,\tilde{z})} \qquad (4.9) \\ N(z,\beta) &= \sum_{\tilde{z}} p(\tilde{z}) e^{-\beta d_{KL}(z,\tilde{z})} \\ p(\tilde{z}) &= \sum_{z} p(\tilde{z} \mid z) p(z) \qquad (4.10) \\ \tilde{z} &= \sum_{z} p(z \mid \tilde{z}) z \qquad (4.11) \end{aligned}$$

A partir de ces équations, il est désormais possible d'utiliser l'algorithme Espérance-Maximisation, pour calculer la courbe débit-distorsion et résoudre le problème du principe IB en donnant un minimum local. Connaissant les liens entre X et Z, les équations consistantes précédentes se réécrivent sous la forme donnée par Tishby et al. (1999) :

$$\begin{aligned} p(\tilde{x} \mid x) &= \frac{p(\tilde{x})}{N(x,\beta)} e^{-\beta \sum_y p(y|x) \log \frac{p(y|x)}{p(y|\tilde{x})}} \qquad (4.12) \\ N(x,\beta) &= \sum_{\tilde{x}} p(\tilde{x}) e^{-\beta \sum_y p(y|x) \log \frac{p(y|x)}{p(y|\tilde{x})}} \\ p(\tilde{x}) &= \sum_{x} p(\tilde{x} \mid x) p(x) \qquad (4.13) \\ p(y \mid \tilde{x}) &= \sum_{x} p(x \mid \tilde{x}) p(y \mid x) \qquad (4.14) \end{aligned}$$

Nous pouvons noter que l'espace de reproduction $\tilde{\mathcal{X}}$ n'est pas calculé et que celui-ci est implicite. En effet, l'espace de reproduction est appréhendé par les dépendances qui le lient à l'information véhiculée par Y. En termes d'indexation, la minimisation de ce principe donne les probabilités d'assignement, c'est-à-dire un clustering flou, où les centroïdes des clusters sont représentés par les distributions conditionnelles $p(y \mid \tilde{x})$.
En prenant les distributions qui vérifient les équations consistantes $p^*(\tilde{x} \mid x)$, $p^*(\tilde{x})$ et $p^*(y \mid \tilde{x})$, la fonction débit-distorsion paramétrique s'écrit comme suit. Tout d'abord, nous donnons la fonction de débit paramétrée par β :

$$\begin{aligned} R(\beta) &= I(X,\tilde{X}) \\ &= \sum_{x,\tilde{x}} p^*(\tilde{x} \mid x) p(x) \log \frac{p^*(\tilde{x} \mid x)}{p^*(\tilde{x})} \qquad (4.15) \end{aligned}$$

La fonction de distorsion s'écrit quant à elle par :

$$\begin{aligned} D(\beta) &= \mathbb{E}_{X,\tilde{X}} \Big[d_{KL}(p(Y \mid X) \mid p(Y \mid \tilde{X})) \Big] \\ &= \sum_{x,\tilde{x}} p^*(\tilde{x} \mid x) p(x) \sum_{y} p(y \mid x) \log \frac{p(y \mid x)}{p^*(y \mid \tilde{x})} \qquad (4.16) \end{aligned}$$

Ces deux quantités représentent une quantité d'information. Aussi quand l'une diminue l'autre augmente et vice versa. Ces deux mesures nous permettent de quantifier l'information qui est conservée ou extraite en rapport à l'information d'importance perdue.

Entrées : l'ensemble des fonctions $\{p(Y \mid x_i)\}_{i=1}^{n}$, la divergence de Bregman D_ϕ, le nombre de groupes k, le paramètre β et la distribution $\{p(x_i)\}_{i=1}^{n}$
Sorties : l'ensemble des fonctions $\{p(Y \mid \tilde{x}_i)\}_{i=1}^{k}$, les probabilités $\{\{p(\hat{x}_i \mid x_j)\}_{i=1}^{k}\}_{j=1}^{n}$ et $\{p(\tilde{x}_i)\}_{i=1}^{k}$
Initialisation : choisir les sorties quelconques telles qu'elles respectent l'intégration à 1.
while convergence **do**
étape de Maximisation
for $j = 1$ to n **do**
for $i = 1$ to k **do**

$$p(\tilde{x}_i \mid x_j) = \frac{p(\tilde{x}_i) e^{-\beta \sum_y p(y|x_j) \log \frac{p(y|x_j)}{p(y|\tilde{x}_i)}}}{\sum_{\tilde{x}} p(\tilde{x}) e^{-\beta d(x_j, \tilde{x})}}$$

end for
end for
for $i = 1$ to k **do**
$p(\tilde{x}_i) = \sum_x p(\tilde{x}_i \mid x) p(x)$
end for
étape d'Espérance
for $i = 1$ to k **do**
$p(Y \mid \tilde{x}_i) = \sum_x p(x \mid \tilde{x}_i) p(Y \mid x)$
end for
end while

Algorithm 2 – Algorithme Espérance-Maximisation pour le calcul de la courbe débit-distorsion liée au principe IB. Cet algorithme permet de calculer un point de la courbe en fonction du paramètre de compromis β.

4.1.3 Algorithme

Nous réécrivons l'algorithme décrit au §3.4.2.2 en considérant les équations consistantes Eq. 4.12, Eq. 4.13 et Eq. 4.14. Cet algorithme permet de calculer les probabilités d'assignement et l'espace de reproduction $\tilde{\mathcal{Z}}$. Nous utilisons comme critère de convergence la différence des sorties à un instant t et $t+1$ de la boucle *while*. Quand cette différence est plus petite qu'un certain seuil prédéfini, nous considérons que l'algorithme a convergé et stoppe. Ce critère de convergence permet d'évaluer si les sorties se stabilisent.
Nous remarquons qu'il n'y a pas un grand changement avec l'algorithme initial, excepté le fait que les distributions conditionnelles $p(Y \mid \tilde{X})$ sont calculées. Ces probabilités permettent de calculer la quantité d'information d'importance extraite en évaluant l'information mutuelle entre $\tilde{X}$ et Y.
Par la suite, nous nous référons à cet algorithme par la notation $\{p(\tilde{X} \mid X), p(\tilde{X}), p(Y \mid \tilde{X})\} = IB(\beta, p(Y \mid X), p(X), k)$ qui prend certaines entrées pour donner les sorties souhaitées.

4.1.4 Recuit simulé

Nous avons précisé dans les paragraphes précédents que l'algorithme itératif d'Espérance-Maximisation permet de converger vers un minimum local du lagrangien. Pour

résoudre ce problème, les méthodes par recuit simulé (Rose, 1998) permettent de converger vers des minimums globaux en passant d'états stables en états stables. Ces algorithmes dépendent d'un paramètre de température T qui passe continûment de l'infini vers la température absolue 0, ce qui fait varier les états du systèmes en suivant les minima globaux d'une fonction d'énergie. Cette idée a été motivée pour la première fois par l'observation de processus de recuit en physique moléculaire. En effet, certains systèmes peuvent être transformés dans leurs états de plus basse énergie en descendant graduellement la température. Rose (1998) montre que le problème de la minimisation débit-distorsion rentre dans le cadre du recuit simulé où la fonction d'énergie est le lagrangien (Eq. 4.3) et le paramètre de température est inversement proportionnel au multiplicateur de Lagrange, tel que $T = 1/\beta$. Ainsi, le paramètre β varie de 0 vers l'infini en faisant varier les états du sytème. La courbe débit-distorsion est calculée complètement en obtenant continûment des minima globaux.
Lors du recuit simulé, la température doit être abaissée avec précaution pour éviter des sauts d'états stables et se retrouver dans des minima locaux. Par exemple, pour la minimisation débit-distorsion, il y a des transitions de phases qui ne doivent pas être ratées. Ces phases sont directement liées au nombre effectif d'éléments de l'espace de reproduction calculé. Et quand la température diminue, la taille effective de l'espace de reproduction augmente. Par conséquent, pour simuler ces transitions d'un état stable associé à β_1 vers un état stable β_2 de plus basse énergie, la différence entre les deux paramètres ne doit pas excéder une certaine limite. Cette limite garantit que la transition n'omet pas de phase, ainsi évitant un minimum local.
Une fonction d'énergie libre est associée à chaque état du système. Dans notre cas, la fonction d'énergie libre $\mathcal{F}$ est définie comme le lagrangien lié à β et calculé pour les probabilités qui vérifient les équations consistantes Eq .4.12 et Eq .4.13. Elle correspond à l'énergie résiduelle d'un état. Aussi pour un β fixé, cette énergie doit être minimale.

$$\begin{aligned}
\mathcal{F} &= R(\beta) + \beta D(\beta) && (4.17)\\
&= \sum_{x,\tilde{x}} p(\tilde{x},x)\log\frac{p(\tilde{x})e^{-\beta d_{KL}(p(Y|x)|p(Y|\tilde{x}))}}{N(x,\beta)p(\tilde{x})} + \beta D(\beta)\\
&= -\sum_{x,\tilde{x}} p(\tilde{x},x)\left[\log N(x,\beta) + \beta d_{KL}(p(Y\mid x)\mid p(Y\mid\tilde{x}))\right] + \beta D(\beta)\\
&= -\sum_{x} p(x)\log N(x,\beta) && (4.18)
\end{aligned}$$

En conséquence, nous souhaitons traquer les énergies libres minimales en même temps que la température est abaissée. Pour commencer, quand le système est dans son état initial ($\beta = 0$), les probabilités d'assignement sont uniformes, et l'espace de reproduction contient un seul élément identifiable qui correspond à la moyenne de l'espace d'entrée. Les k éléments de l'espace de reproduction fusionnent en un seul élément. Ainsi, nous nous retrouvons avec une information mutuelle $I(X,\tilde{X})$ nulle et une distorsion maximale. Quand β croît, certains éléments de l'espace de reproduction se séparent et se différencient. Cette séparation correspond à une transition de phase. Pour conserver l'énergie libre du système, l'espace de reproduction est modifié tel que :

$$\frac{\partial F}{\partial \tilde{\mathcal{X}}} = 0 \qquad (4.19)$$

La transition ne fait pas varier l'énergie libre du système. Cette condition est équivalente à l'équation consistante Eq. 3.97. D'autre part, le Hessien de cette fonction d'énergie doit

être défini positif pour toutes perturbations du système ou de l'espace de reproduction. Cette condition implique que la fonction a atteint un minimum et non une forme de selle. Nous obtenons la condition suivante :

$$\mathcal{H}(\mathcal{F}) = \frac{\partial^2 F}{\partial \tilde{\mathcal{X}}^2} \geq 0 \tag{4.20}$$

Les transitions de phases apparaissent quand l'égalité est atteinte dans Eq. 4.20. Par conséquent, quand le système est dans un état stable donné, l'inégalité précédente nous donne une borne sur l'augmentation du paramètre de compromis β. Ainsi en limitant l'augmentation de β, nous contrôlons la transition de phase avant de converger vers le nouvel état stable suivant. Dans le cas du principe IB, cette limite n'a pas été calculée et nous donnons ici les équations qui permettent de calculer le multiplicateur de Lagrange critique pour chaque état stable.
Premièrement, nous donnons l'expression du Hessien de la fonction d'énergie libre. Pour rendre le calcul plus pratique, nous effectuerons les calculs avec les notations utilisant Z et $\tilde{Z}$ tel que la fonction d'énergie libre s'écrit :

$$\mathcal{F} = -\sum_z p(z) \log N(z, \beta) \tag{4.21}$$

Nous donnons pour commencer le gradient de l'énergie libre par rapport à un $\tilde{z}$ particulier :

$$\nabla_{\tilde{z}} \mathcal{F} = \beta \sum_z p(z) p(\tilde{z} \mid z) \nabla_{\tilde{z}} d_{KL}(z, \tilde{z}) \tag{4.22}$$

où le gradient de la divergence de Kullback-Leibler est donné par l'équation suivante en indexant les vecteurs $z = [z_1, ...z_n]^T$ et $\tilde{z} = [\tilde{z}_1, ...\tilde{z}_n]^T$:

$$\nabla_{\tilde{z}} d_{KL}(z, \tilde{z}) = \left[-\frac{z_1}{\tilde{z}_1}, ... - \frac{z_n}{\tilde{z}_n} \right]^T \tag{4.23}$$

Pour calculer les éléments du Hessien, nous considérons deux réalisations différentes de $\tilde{Z}$ que nous notons $\tilde{z}^i$ et $\tilde{z}^j$. Alors nous pouvons calculer la dérivée partielle suivante nécessaire au calcul du Hessien :

$$\frac{\partial^2 \mathcal{F}}{\partial \tilde{z}^j \partial \tilde{z}^i} = \beta^2 \sum_z p(z) p(\tilde{z}^i \mid z) p(\tilde{z}^j \mid z) (\nabla_{\tilde{z}^j} d_{KL}(z, \tilde{z}^j) \nabla_{\tilde{z}^i} d_{KL}(z, \tilde{z}^i)^T) \tag{4.24}$$

Cette dérivée partielle est en fait une matrice carrée de taille $n \times n$ en rapport avec la taille des vecteurs $\tilde{z}$. Dans le cas où $\tilde{z}^j = \tilde{z}^i = \tilde{z}$, c'est à dire que la dérivée seconde est calculée par rapport à la même variable, celle-ci s'exprime par :

$$\begin{aligned} \frac{\partial^2 \mathcal{F}}{\partial \tilde{z}^2} &= \beta^2 \sum_z p(z) p(\tilde{z} \mid z)^2 (\nabla_{\tilde{z}} d_{KL}(z, \tilde{z}) \nabla_{\tilde{z}} d_{KL}(z, \tilde{z})^T) \\ &+ \beta \sum_z p(z) p(\tilde{z} \mid z) \{ \frac{\partial^2 d_{KL}(z, \tilde{z})}{\partial \tilde{z}^2} - \beta (\nabla_{\tilde{z}} d_{KL}(z, \tilde{z}) \nabla_{\tilde{z}^i} d_{KL}(z, \tilde{z}^i)^T) \} \end{aligned} \tag{4.25}$$

Cette dérivée seconde est aussi une matrice carrée de taille $n \times n$. Alors, le Hessien qui est une matrice formée des dérivées partielles, est constitué de matrices carrées. Si k est le nombre d'éléments de l'ensemble $\tilde{\mathcal{Z}}$, alors le Hessien s'écrit :

$$\mathcal{H}(\mathcal{F}) = \left\{ \frac{\partial^2 \mathcal{F}}{\partial \tilde{z}^j \partial \tilde{z}^i} \right\}_{1 \leq i,j \leq k} \tag{4.26}$$

Cette matrice est symétrique par construction et elle est de taille $kn \times kn$ car chacun de ses éléments est une matrice carrée. Nous développons, dans le cas de la divergence de Kullback-Leibler, les matrices représentant ses dérivées secondes partielles :

$$\nabla_{\tilde{z}^j} d_{KL}(z, \tilde{z}^j) \nabla_{\tilde{z}^i} d_{KL}(z, \tilde{z}^i)^T = \{\frac{z_l z_m}{\tilde{z}_l^j \tilde{z}_m^i}\}_{1 \leq l,m \leq n} \quad (4.27)$$

$$\frac{\partial^2 d_{KL}(z, \tilde{z})}{\partial \tilde{z}^2} = \text{diag}\{\frac{z_i}{\tilde{z}_i^2}\}_{1 \leq i \leq n} \quad (4.28)$$

Il suffit de remplacer les matrices précédentes dans les équations Eq. 4.24 et Eq. 4.25 pour calculer les dérivées partielles. A partir de cette définition du Hessien, nous voulons maintenant déterminer un β critique où une transition se produit. Un β critique implique que $\det(\mathcal{H}) = 0$. Pour déterminer les paramètres qui satisfont cette condition, nous introduisons deux matrices A, B pour décomposer $\mathcal{H}$. Comme dans Eq. 4.26, nous notons $A = \{a_{i,j}\}_{1 \leq i,j \leq k}$ et $B = \{b_{i,j}\}_{1 \leq i,j \leq k}$. Les éléments de A sont définis par :

$$a_{i,j} = \begin{cases} \sum_z p(z) p(\tilde{z}^i \mid z) p(\tilde{z}^j \mid z) (\nabla_{\tilde{z}^j} d_{KL}(z, \tilde{z}^j) \nabla_{\tilde{z}^i} d_{KL}(z, \tilde{z}^i)^T) & \text{, si } i \neq j \\ \sum_z p(z) p(\tilde{z}^i \mid z) (p(\tilde{z}^i \mid z) - 1) (\nabla_{\tilde{z}^i} d_{KL}(z, \tilde{z}^i) \nabla_{\tilde{z}^i} d_{KL}(z, \tilde{z}^i)^T) & \text{, si } i = j \end{cases} \quad (4.29)$$

Les éléments de B sont quant à eux définis par :

$$b_{i,j} = \begin{cases} 0 & \text{, si } i \neq j \\ \sum_z p(z) p(\tilde{z} \mid z) \frac{\partial^2 d_{KL}(z, \tilde{z})}{\partial \tilde{z}^2} & \text{, si } i = j \end{cases} \quad (4.30)$$

Cette dernière matrice B est diagonale puisqu'elle est composée de sous matrices diagonales définies dans Eq. 4.28. Ces deux matrices permettent de redéfinir le Hessien sous la forme suivante :

$$\mathcal{H}(\mathcal{F}) = \beta^2 A + \beta B \quad (4.31)$$

Comme B est diagonale d'éléments non nuls, elle est inversible. Nous pouvons alors donner l'égalité suivante sur le déterminant du Hessien :

$$\det(\mathcal{H}(\mathcal{F})) = \beta^{2nk} \det(B) \det(B^{-1} A + \frac{Id}{\beta}) \quad (4.32)$$

Ainsi pour trouver le paramètre critique β_c en supposant que celui-ci est non nul, il suffit de résoudre l'équation :

$$\det(B^{-1} A + \frac{Id}{\beta}) = 0 \quad (4.33)$$

Nous observons que $-1/\beta_c$ fait partie des valeurs propres de la matrice $B^{-1}A$. Pour ne manquer aucune transition de phases, nous prenons β_c tel que la différence entre β et β_c soit la plus petite. Cela revient à prendre $\beta_c = -1/\lambda_{max}$ où λ_{max} est la plus grande valeur propre en valeur absolue.
En conclusion, à partir d'un état stable lié à un paramètre β, nous avons une formulation analytique pour calculer le paramètre de compromis suivant β_c et ainsi changer de phase. Ensuite le système se stabilise dans ce nouvel état lié à β_c. Cette procédure permet de converger vers des minima globaux pour chaque état. Nous décrivons par la suite l'algorithme qui découle du recuit simulé.

4.1.5 Algorithme de recuit simulé

Nous décrivons comment l'algorithme du recuit simulé peut être défini à partir des équations de transitions de phases précédentes. Le but est de passer d'états stables en états stables en n'omettant aucune transition de phase. A chaque fois, qu'une transition de phase est atteinte, il faut laisser le système revenir dans un état stable. Ainsi, l'algorithme est une succession de transitions et de retours à l'équilibre. Ainsi, l'algorithme est décrit par l'Algorithm 3. Cet algorithme nous permet d'échantillonner la courbe débit-

Entrées : l'ensemble de fonctions $\{p(Y \mid x_i)\}_{i=1}^{n}$, le nombre de groupe k et la distribution $\{p(x_i)\}_{i=1}^{n}$
Sorties : la fonction débit-distorsion et les paramètres de chaque état
Initialisation : prendre $\beta = 0$ et intialiser les sorties de IB
while $D \neq 0$ **do**
- $\beta = \beta_c$ (cf. Eq. 4.33)
- initialiser l'algorithme IB avec les sorties précédentes $\{p(\tilde{X} \mid X), p(\tilde{X}), p(Y \mid \tilde{X})\}$
- lancer IB pour obtenir l'equilibre :$\{p(\tilde{X} \mid X), p(\tilde{X}), p(Y \mid \tilde{X})\} = IB(\beta, p(Y \mid X), p(X), k)$
- calculer $R(\beta)$ et $D(\beta)$

end while

Algorithm 3 – L'algorithme de recuit simulé permet de passer d'états stables en états stables en augmentant le paramètre β en suivant chaque transition de phases. Il permet de converger vers des minima globaux de la fonction d'énergie libre.

distorsion liée au principe IB en passant de minimum global en minimum global. Nous verrons par la suite que cette courbe débit-distorsion nous servira à déterminer la quantité d'information à extraire.
Geman & Geman (1984) décrivent une procédure de convergence globale un peu différente de celle proposée. Ils considèrent que β augmente exponentiellement. Dans l'algorithme du recuit simulé la procédure $\beta = \beta_c$ devient $\beta = \alpha\beta$ avec $\alpha > 1$. Nous montrerons que cette procédure fonctionne aussi bien que le recuit simulé dans le cas du principe IB en comparant les courbes débit-distorsion. Nous faisons la comparaison puisque cette dernière procédure est moins complexe, effectuant un calcul très simple.

4.2 Caractérisation de l'information d'importance

Dans cette partie, nous nous intéressons à définir l'information d'importance qui est modélisée par la variable aléatoire Y. Nous donnons ensuite un critère heuristique pour déterminer la quantité d'information d'importance à extraire.

4.2.1 Choix de l'information pertinente

Nous avons vu au §3.5.3, que dans une décomposition de l'information en deux parties, l'information pertinente est contenue dans le modèle. En effet, cette information pertinente est en rapport direct avec le pouvoir du modèle à générer les objets. Prenons un exemple pour clarifier cette idée. Considérons deux chaînes infinies construites sur l'alphabet $\{0, 1\}$ dont une est complètement aléatoire et l'autre suit la forme $01010101010101\ldots$. Il est clair que la propriété de la première chaîne est d'être aléatoire et la propriété de la

deuxième est d'être formée d'une suite de 01. Les modèles qui peuvent être construits pour représenter ces deux chaînes renferment ces propriétés. Par exemple, la chaîne aléatoire admettra comme meilleur modèle probabiliste, un modèle de source sans mémoire tel que $p(0) = p(1) = 1/2$. Si jamais nous essayons de modéliser cette chaîne aléatoire infinie par un modèle markovien de premier ordre, alors la longueur de code de la chaîne sachant le modèle sera équivalente et le codage du modèle nécessitera plus de bits. En effet, pour modéliser un champ markovien sur $\{0, 1\}$, il faut 2 paramètres en plus par rapport à une source sans mémoire. En conséquence, la longueur de code totale sera supérieure au cas précédent ce qui va à l'encontre du principe de longueur de description minimum. Pour l'autre chaîne $0101010101...$, il est évident qu'un modèle de Markov $p(0 \mid 1) = p(1 \mid 0) = 1$ est optimal en termes de codage. En effet, ce modèle dit que les bits changent à chaque fois. La longueur minimale moyenne est $\lim_{n\to\infty} \frac{1}{n} + \frac{3}{2n} \log n$ où 3 paramètres sont nécessaires à décrire le modèle. Si toutefois, nous essayons de modéliser cette chaîne avec une probabilité de source sans mémoire, nous obtenons une longueur moyenne de code égale à $\lim_{n\to\infty} 1 + \frac{1}{2n} \log n$ où un paramètre est nécessaire à déterminer la probabilité de la source $p(0) = p(1) = 1/2$. Il est évident que ce modèle est bien pire que le modèle de Markov. Nous remarquons, qu'avec le même modèle de source sans mémoire, les deux chaînes exhibent les mêmes propriétés. Néanmoins, dans l'autre cas en utilisant le même modèle markovien, les chaînes n'auront plus les mêmes propriétés. Considérons maintenant un modèle plus complexe pour modéliser la suite $01010101010101....$ Par exemple, cette chaîne peut être formée d'une succession d'éléments du type 01 ou 0101 et ainsi de suite. Est-il possible de gagner en longueur de codage ? Notons, l la longueur des mots 01 ou 0101, etc. Alors, l'alphabet est maintenant de taille 2^l. Considérons un modèle markovien d'ordre 1. Le nombre de paramètres nécessaires à représenter les probabilités de transitions et la probabilité initiale est $(2^l - 1)2^l+1$. Ainsi la longueur de code moyenne devient $\lim_{n\to\infty} \frac{l}{2^l n} + \frac{(2^l-1)2^l+1}{2n}(\log n - \log l)$, où n est un multiple de l. Nous remarquons que la longueur $\frac{l}{2^l n}$ diminue beaucoup quand l grandit. Néanmoins, le terme dominant de la longueur de code est $\frac{(2^l-1)2^l+1}{2n} \log n$ et il augmente avec la nombre l. Aussi, le meilleur cas est obtenu pour $l = 1$. En l'occurence, nous avons essayé d'ajouter de nouvelles propriétés au signal au travers de modèles plus complexes, mais le principe LDM se concentrera sur les propriétés essentielles du signal. Nous remarquons d'ailleurs que prendre $l = 1$ et $l = 2$ engendre le même type de propriétés. Ces quelques exemples montrent bien en quoi les modèles contiennent l'information d'importance contenue dans les signaux.

Considérons un ensemble de modèles statistiques $\{\mathcal{M}_i\}$ de la variable aléatoire X. En outre, chaque modèle représente une fonction de probabilité de la source $p(X \mid \mathcal{M}_i)$. Sur cet espace de modèles, nous définissons une variable aléatoire $\mathcal{M}$ qui prend ses valeurs dans cet espace en suivant un certaine distribution $w(\mathcal{M})$. En combinant cette modélisation avec le fait que les modèles contiennent l'information pertinente, nous pouvons considérer que l'information d'importance ou d'intérêt est portée par l'ensemble des modèles. Alors, nous prenons la variable Y telle que $Y = \mathcal{M}$. De plus, nous avons vu précédemment qu'un modèle fixé fait ressortir un certain type d'information des signaux analysés. Pour certains signaux (par exemple la chaîne aléatoire) toute l'information d'intérêt est extraite tandis que pour d'autre cas (par exemple la chaîne $0101...$) aucune information d'importance n'est extraite. Ainsi, en comparant les signaux par rapport aux modèles nous pouvons sélectionner l'information d'importance nécessaire à la discrimination. Ces considérations nous amènent à définir le principe d'*Information Bot-*

tleneck suivant :

$$\min_{p(\tilde{x}|x),p(\mathcal{M}_i|\tilde{x})} I(\tilde{X}, X) - \beta I(\tilde{X}, \mathcal{M}) \tag{4.34}$$

En appliquant l'algorithme IB, nous pouvons obtenir les distributions conditionnelles $p(\mathcal{M} \mid \tilde{X})$ qui sont les prototypes des groupes. Ces distributions représentent les types d'information qui sont représentatifs du groupe. En considérant plusieurs modèles, nous fusionnons plusieurs types d'information. Nous montrons par la suite que ce critère inclut une sélection de modèles.

4.2.2 Liens avec la redondance de codage

Nous soulignons le fait que le principe IB se rapporte à la sélection de modèle en établissant un lien avec le problème minimax de la redondance de codage (cf. §3.3.4). Nous étendons le problème minimax au cas où les paramètres sont donnés par les réalisations de $\mathcal{M}$. D'autre part, Rissanen (n.d.) montre que le problème minimax de la redondance de codage est un cas particulier du problème plus général suivant :

$$\min_{q(X)} \max_{w(\mathcal{M})} \sum_i w(\mathcal{M}_i) R(\mathcal{M}_i, q) \tag{4.35}$$

Le problème minimax devient un cas particulier du problème précédent en restreignant la probabilité $w(\mathcal{M})$ à être déterministe. En fin de compte, ce critère revient à évaluer l'information mutuelle entre les données X et l'ensemble de modèles $\mathcal{M}$. Par conséquent, le problème minimax se réécrit comme :

$$\min_{q(X)} \max_{w(\mathcal{M})} I(X, \mathcal{M}) \tag{4.36}$$

Si nous avons un canal bidirectionnel entre X et $\mathcal{M}$, d'un coté nous essayons de minimiser la capacité du canal de X vers $\mathcal{M}$ tout en maximisant la capacité dans le sens inverse. Si la probabilité $w(\mathcal{M})$ est fixée, alors la solution optimale est :

$$q^*(x) = \sum_i p(x \mid \mathcal{M}_i) w(\mathcal{M}_i) \tag{4.37}$$

La distribution universelle $q^*(X)$ est donnée par un mélange d'informations correspondant aux différents modèles. Cette information est toujours divisée en deux parties.
Revenons au principe IB (Eq. 4.34) dans le cas où l'information d'importance est portée par $\mathcal{M}$. Nous supposons connaître les distributions $p(X)$ et $p(M \mid X)$. Quand la fonctionnelle est minimisée globalement, nous maximisons en particulier $I(\tilde{X}, \mathcal{M})$ en fonction de $p(\mathcal{M} \mid \tilde{X})$. Si $p(\tilde{X})$ est fixée, cela revient à maximiser en fonction de $w(\mathcal{M}) = \sum_{\tilde{x}} p(\mathcal{M} \mid \tilde{x}) p(\tilde{x})$. En relation avec le problème minimax, nous maximisons la redondance de codage de $\tilde{X}$ pour la distribution $p(\tilde{X})$ fixée. Cependant cette dernière distribution n'est pas fixée et elle est liée à la minimisation de $I(X, \tilde{X})$. Comme la minimisation se fait sur $p(\tilde{X} \mid X)$, cela revient à trouver $p(\tilde{X})$ telle que la longueur de code $I(X, \tilde{X})$ nécessaire à coder $\tilde{X}$ soit la plus petite possible. Par conséquent, le principe d'*Information Bottleneck* essaie d'extraire de $p(X)$ une distribution $p(\tilde{X})$ qui serait un code universel pour tous processus générés par n'importe quel modèle. Par le principe IB, nous voulons avoir un codage universel d'une partie de l'information contenue dans X vis à vis de l'ensemble des modèles. Par conséquent, nous déduisons que ce principe inclut une sélection de modèles. Nous rappelons que le principe IB induit un clustering flou de l'espace d'entrée.

Ainsi, chaque groupe privilégie tel ou tel modèle par la probabilité $p(\mathcal{M} \mid \tilde{X})$; c'est une sélection floue des modèles.
En conclusion, nous avons introduit un critère qui permet de faire un sélection floue de modèles pour extraire l'information d'intérêt contenue dans une variable X. Ce critère est fondé sur la combinaison du principe IB avec une variable aléatoire d'intérêt prenant ses valeurs dans un espace de modèles. Comme, nous l'avons noté ce principe permet de régler le compromis entre la quantité d'information extraite et la quantité d'information d'importance.

4.2.3 Détermination de la quantité d'information d'importance

La méthodologie décrite jusqu'à présent permet de calculer une fonction débit-distorsion où la quantité d'information d'intérêt est mesurée. Cependant, pour indexer un volume de données, nous sommes intéressés par un seul point de cette courbe débit-distorsion qui nous donne la quantité d'information extraite et la quantité d'information d'importance. Par conséquent, nous souhaitons obtenir un critère objectif sur la fonction débit-distorsion pour déterminer le compromis optimal.
Tout d'abord, nous décrivons un critère heuristique introduit par Sugar & James (1998) pour déterminer le nombre optimal de clusters dans une procédure de clustering du type k-moyennes. Ce type d'algorithme agit sur des données observables d'un espace vectoriel de dimension fini. Nous notons ces variables $[x_1, ...x_n]$, où n est le nombre de réalisations. Alors la problématique des k-moyennes consiste à trouver les k centroïdes $[\hat{x}_1, ...\hat{x}_k]$ et la fonction d'assignement déterministe $p(\hat{X} \mid X)$ qui minimise la mesure de distorsion suivante :

$$D(k) = \min_{[\hat{x}_1,...\hat{x}_k]} \mathbb{E}_{X,\hat{X}} \left[(X - \hat{X})^T (X - \hat{X}) \right] \tag{4.38}$$

Sugar & James (1998) donnent le critère heuristique suivant pour déterminer le nombre optimal de clusters :

$$\hat{k} = \arg\max_k \{D(k+1)^{-n/2} - D(k)^{-n/2}\} \tag{4.39}$$

Le but est de localiser des sauts dans la fonction de distorsion en fonction de k. Cette idée vient de la constatation que les variations de la distorsion changent brutalement quand k approche du nombre optimal de clusters. Ils montrent en particulier des résultats asymptotiques pour des données générées par un mélange de g gaussiennes de même variance et dont les centres sont suffisamment éloignés tels que g groupes se distinguent :

$$D(k)^{-n/2} \approx \begin{cases} \frac{k}{g} & \text{si } k \leq g \\ 0 & \text{si } k > g \end{cases} \tag{4.40}$$

Nous constatons que les tendances changent en fonction du nombre de clusters k. Quand k est supérieur à g, les gains en distorsion sont faibles. En d'autres termes, les g groupes optimaux sont subdivisés en sous-groupes. Au contraire, quand k est inférieur à g, les gains en distorsion sont conséquents, c'est-à-dire que nous gagnons à nous approcher du nombre optimal de groupes.
Nous avons noté au §3.4.2.2 que l'algorithme des k-moyennes est un problème de débit-distorsion, dont le débit est donné par $I(X, \hat{X})$, avec un probabilité d'assignement déterministe. Dans la cas d'un clustering par k-moyennes, cette quantité d'information peut

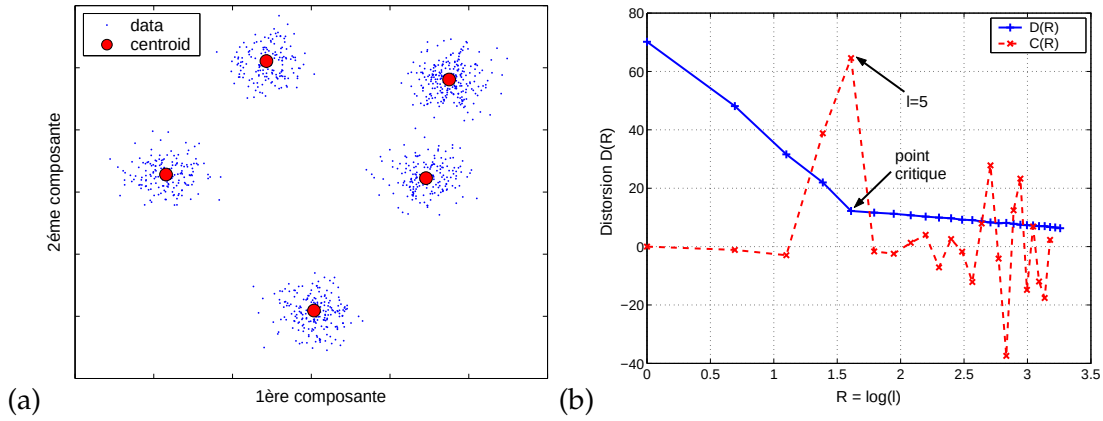


FIG. 4.2 – (a) Données générées à partir d'un mélange de 5 gaussiennes de même variance et dont les centres sont éloignés. (b) Courbe débit-distorsion correspondant au clustering par k-moyennes.

être approximée par $I(X, \hat{X}) = H(\hat{X})$ (cf. Eq. 3.88). Si les réalisations sont équitablement distribuées dans chaque groupe, une approximation au premier ordre de $H(\hat{X})$ est donnée par $\log k$ où k est le nombre de clusters. Comme le débit est directement lié au nombre de clusters, nous pouvons nous attendre à observer les changements de variations dans la courbe débit-distorsion. A titre d'exemple, nous donnons la courbe débit-distorsion correspondant au clustering par k-moyennes des données présentées dans la Figure 4.2.(a). Ces données ont été générées par un mélange de 5 gaussiennes de même variance. Nous observons dans la Figure 4.2.(b) un changement abrupt du comportement de la courbe $D(R)$ pour un débit obtenu avec 5 clusters (le nombre optimal de clusters). Quand le nombre de groupes est plus petit que l'optimal, la distorsion varie beaucoup avec le débit. Au contraire quand le nombre de clusters est plus grand que l'optimal, la distorsion varie peu avec le débit. Alors, nous proposons un critère qui vise à trouver ce point de transition. Ce point est obtenu pour une courbure maximale de la fonction débit-distorsion. Ce critère s'exprime dans le cas d'une fonction paramétrique par :

$$\hat{\beta} = \arg\sup_{\beta} \frac{|R'(\beta)D''(\beta) - R''(\beta)D'(\beta)|}{(R'(\beta)^2 + D'(\beta)^2)^{\frac{3}{2}}} \tag{4.41}$$

Nous sélectionnons le paramètre $\hat{\beta}$ qui donne la courbure maximale C de la fonction débit-distorsion. Il est à oter que ce type de critère est utilisé dans les problèmes inverses pour la détermination du paramètre de compromis entre le terme d'attache aux données et le terme de régularisation (Hansen & O'Leary, 1993). Cependant, ce critère nommé courbe en L est défini dans un espace transformé en $\log - \log$ qui n'est pas nécessaire dans notre cas. La fonction de courbure peut encore s'écrire sous la forme suivante :

$$C(R) = \frac{|D''(R)|}{(1 + D'(R)^2)^{3/2}} \tag{4.42}$$

Cette fonction est représentée dans la Figure 4.2.(b), et nous notons qu'elle atteint son maximum au point critique donnant ainsi le nombre de clusters optimal. Nous donnons un second exemple en utilisant cette fois-ci l'algorithme de clustering k-moyennes floues qui est en adéquation avec la théorie débit-distorsion. Le jeu de données utilisé est généré par un mélange de 10 gaussiennes. Nous montrons la courbe débit-distorsion et

sa courbure dans la Figure 4.3.(b). Nous observons que le maximum de la courbure est atteint pour le débit qui correspond à 10 centroïdes identifiables. Le résultat du clustering correspondant au point critique de la courbe est représenté dans la Figure 4.3.(c). D'autre part, nous remarquons un autre point d'intérêt où un maximum local de la fonction $C(R)$ apparaît. Le résultat du clustering associé à ce couple débit-distorsion est présenté dans la Figure 4.3.(d). Nous remarquons que les groupes formés correspondent à une hiérarchisation du clustering optimal. En effet, les données sont groupées en 4 premiers groupes qui sont ensuite divisés en 10 groupes optimaux. Cet aspect n'est pas discuté dans la problèmatique des courbes en L. Les autres pics de la fonction $C(R)$ ne correspondent pas à des points critiques. Ils sont dûs au fait que la courbe $D(R)$ calculée n'est pas convexe, en dépit du fait que celle-ci doit être convexe théoriquement (Cover & Thomas, 1991).
Par ces expérimentations, nous montrons que notre critère heuristique est valide pour déterminer le nombre optimal de clusters ou le point critique d'une courbe débit-distorsion. D'autre part nous remarquons que ce critère peut permettre de hiérarchiser les clusters en considérant les maximums locaux de la fonction de courbure. Néanmoins, le défaut de cette technique réside dans le calcul de la courbure. En effet, nous calculons cette fonction par une approximation directe sans lissage. Nous notons que sur les deux exemples traités, la courbure oscille après le point critique. Cet effet est dû aux non convexités locales de la fonction débit-distorsion. En conclusion, le critère heuristique présenté permet de déterminer le point critique de n'importe quelle courbe débit-distorsion et en particulier les courbes induites par le principe IB. Ce point critique détermine la quantité d'information nécessaire à discriminer les objets en rapport avec l'information d'importance.

4.3 Caractérisation des structures spatio-temporelles

Nous présentons dans ce paragraphe les modèles qui représentent l'information d'importance contenue dans les STIS. Les STIS sont des données où les textures spatiales évoluent au cours du temps, mais où les objets restent majoritairement stables spatialement. Par conséquent, nous sommes intéressés par la modélisation des interactions spatio-temporelles qui existent entre les pixels de la STIS. Nous supposons que l'information d'importance est portée par des modèles de champ aléatoires (Gueguen & Datcu, 2007a). Cela permet d'appliquer le principe décrit à l'Eq. 4.34. Les expériences réalisées sur la STIS sont proposées au §6.1.3.

4.3.1 Les champs aléatoires comme modèles

Nous avons fait remarqué auparavant (cf. §2.2.2) que les champs aléatoires de Gibbs-Markov sont adaptés à la modélisation des textures. Par exemple, ces modèles engendrent peu de paramètres qui discriminent bien les classes de textures. Nous extrapolons ces modèles pour la modélisation des textures en 3 dimensions. Nous espérons ainsi pouvoir discriminer les structures spatio-temporelles en observant l'évolution des textures spatiales.
Nous avons introduit au §3.1.5 deux modèles de champs aléatoires pour la modélisation des interactions d'un signal en 3 dimensions. Tout d'abord, nous avons présenté les champs aléatoires de Gauss-Markov (cf. §3.1.5.1). Nous avons noté que ce modèle dépend du voisinage employé en considérant des interactions à plus ou moins courtes distances. Ce type de modèle trouve son équivalent dans les modèles linéaires en introduisant un

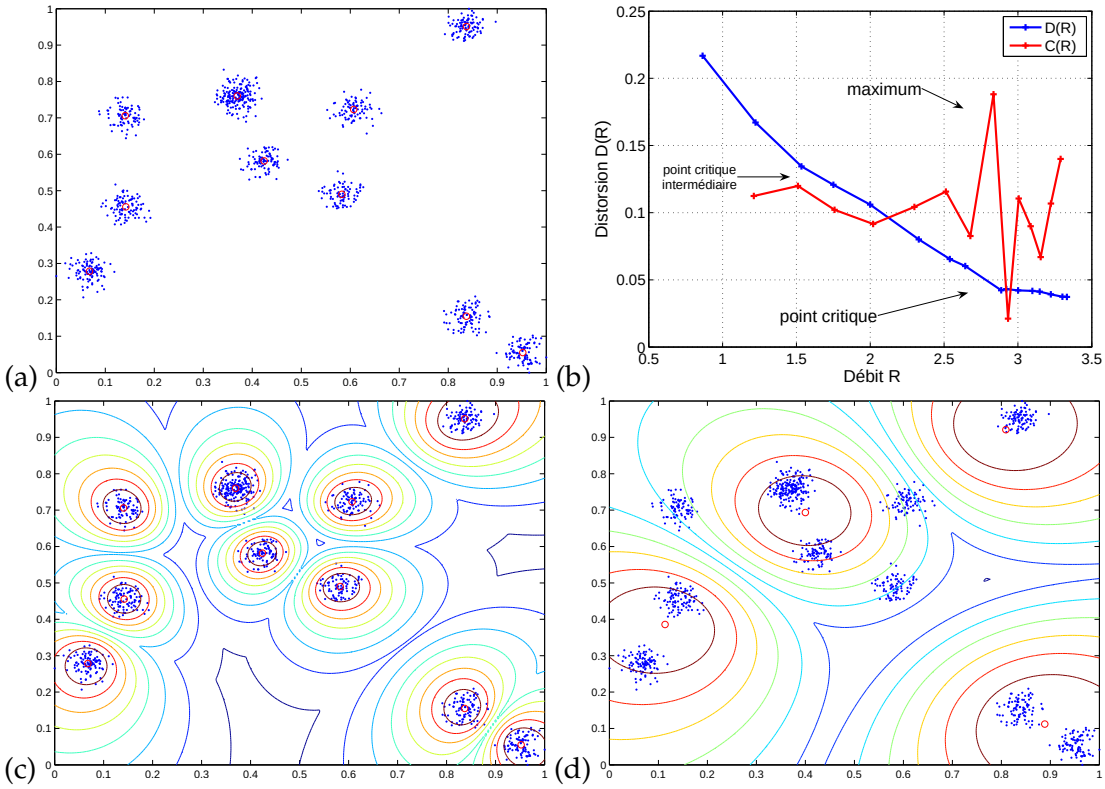


FIG. 4.3 – La figure présente les résultats obtenus avec l'algorithme k-moyennes floues en respectant le critère heuristique Eq. 4.41. (a) Les données générées avec un mélange de 10 gausiennes. (b) La courbe débit-distorsion lié à l'algorithme k-moyennes floues et la courbure $C(R)$. Le maximum de la courbure donne le point de $D(R)$correspond au clustering de (c). Enfin (d) présente le clustering associé au point critique intermédiaire qui donne une hiérarchisation des groupes.

réarrangement des données tel que cela est décrit dans l'Eq. 3.26 où la taille de la transformation linéaire dépend directement du type de voisinage employé. D'autre part, nous avons vu que les paramètres d'un tel modèle s'estiment par Maximum de Vraisemblance en suivant les équations Eq. 3.61 et Eq. 3.62. Ainsi, nous obtenons $\hat{\theta}$ qui modélise les interactions et $\hat{\sigma}$ qui montre à quel point le modèle s'ajuste aux données. Enfin, nous signalons que les différents types de voisinage donnent autant de familles de modèles paramétriques. Aussi, nous pouvons exploiter l'évidence de modèle pour sélectionner le meilleur modèle. Notons par exemple $\mathcal{M}_{\mathcal{N}}$ le modèle associé au voisinage $\mathcal{N}$. Alors, l'évidence de ce modèle calculée pour une réalisation x du champ aléatoire est donnée par :

$$p(x \mid \mathcal{M}_{\mathcal{N}}) = p(x|G, \mathcal{N}(0, .)) \tag{4.43}$$

où la seconde distribution est donnée par Eq. 3.63 et G est calculé suivant les réarrangements induits par le voisinage (cf. §3.1.5.1). Si, nous considérons une variable aléatoire $\mathcal{M}$ qui prend ses valeurs sur l'espace $\{\mathcal{M}_{\mathcal{N}}\}$, le calcul de l'évidence permet de déduire la probabilité conditionnelle $p(X \mid \mathcal{M})$. Ainsi, cette probabilité permettra d'évaluer l'information partagée par les données et les modèles. Enfin, nous pourrons appliquer le principe IB énoncé précédemment.
Nous avons aussi présenté les champs aléatoires autobinomiaux (cf. §3.1.5.2). Contrairement aux champs de Gauss-Markov, ces modèles ont la particularité de modéliser des champs dont les sites prennent leur valeur sur un espace discret fini. Par exemple, les STIS sont des champs dont les pixels prennent leur valeur sur l'intervalle d'entiers $[0, \cdots, 1000]$. Alors cette modélisation est plus adaptée aux STIS que les champs de Gauss-Markov. Cependant l'inférence des paramètres reste incommensurable et nécessite de ce fait quelques approximations. Ces approximations, rappelées par (Schroder et al., 1998), permettent de représenter les champs autobinomiaux par des modèles linéaires. Nous rappelons les expressions de la moyenne et de la variance conditionnées dans Eq. 3.31 et Eq. 3.32. Pour estimer les paramètres, nous employons un estimateur des moindres carrés conditionné en supposant l'approximation de la pseudo-vraisemblance (cf. Eq. 3.21) :

$$\hat{\theta} = \arg\min_{\theta} \sum_{s} (x_s - \mathbb{E}_{X_s|N_s,\theta}[X_s])^2 \tag{4.44}$$

$$= \arg\min_{\theta} \sum_{s} (x_s - \frac{\mathcal{G}}{1 + e^{-\eta}})^2 \tag{4.45}$$

Cette expression est approximée au premier ordre par les séries de Taylor :

$$\eta \approx -\log(\frac{\mathcal{G}}{x_s} - 1) + e_s \tag{4.46}$$

où e_s est un bruit blanc gaussien de moyenne nulle et de petite variance. En remplaçant η dans Eq. 4.46 par sa formulation exacte donnée dans Eq. 3.30, nous obtenons le système linéaire suivant :

$$\vec{d} = G\vec{\theta} - \vec{e} \tag{4.47}$$

avec $\vec{\theta}$ le vecteur de paramètres, $\vec{d}$ un vecteur transformé, $\vec{e}$ un vecteur de bruit blanc gaussien de moyenne nulle et G la matrice des pixels voisins. Le vecteur $\vec{d}$ et la matrice G sont définis par les équations suivantes en utilisant les réarrangements de positions donnés au §3.1.5.1 :

$$\vec{d}_{a(s)} = -\log(\frac{\mathcal{G}}{x_s} - 1) \tag{4.48}$$

$$G_{a(s),b(r)} = \begin{cases} 1 & \text{, if } r = 0 \\ \frac{X_{s+r}+X_{s-r}}{\mathcal{G}} & \text{, if } r \in N \end{cases} \tag{4.49}$$

Comme le système approximé est linéaire, nous utilisons les mêmes méthodes (cf. §3.1.5.1) pour calculer les paramètres et l'évidence du modèle. Nous pouvons ainsi définir l'évidence des modèles autobinomiaux. Même si ces modèles ressemblent aux champs de Gauss-Markov, les champs autobinomiaux couvrent une plus large classe de textures. En conclusion, nous avons plusieurs familles de modèles paramétriques qui sont les champs aléatoires de Gauss-Markov et autobinomiaux et qui représentent l'information d'importance contenue dans les STIS.

4.3.2 Calcul des caractéristiques du canal de communication

Nous considérons que la variable aléatoire $\mathcal{M}$ prend ses valeurs dans l'ensemble des modèles formé par les modèles autobinomiaux des trois premiers ordres et les champs de Gauss-Markov des trois premiers ordres. Nous notons ces modèles par $\mathcal{M}_i$. Pour appliquer le principe IB dans le but de regrouper les réalisations d'un champ aléatoire X en fonction de l'information d'importance contenue dans $\mathcal{M}$, il nous faut calculer les distributions $p(X)$ et $p(\mathcal{M} \mid X)$.
Supposons que nous connaissons les distributions $p(\mathcal{M})$ et $p(X \mid \mathcal{M})$ a priori. La seconde probabilité correspond à l'évidence des modèles et peut être approximée par l'Eq. 3.63 pour chaque réalisation de modèle dans notre cas. Ainsi pour chaque modèle et chaque réalisation x, nous obtenons :

$$p(\mathcal{M}_i \mid x) = \frac{p(x \mid \mathcal{M}_i)p(\mathcal{M}_i)}{\sum_i p(x \mid \mathcal{M}_i)p(\mathcal{M}_i)} \tag{4.50}$$

Cette distribution conditionnelle nous permet d'évaluer quel modèle est le plus à même de représenter la réalisation x. Nous proposons d'estimer la probabilité des modèles par une procédure de sélection de modèle. Ainsi, les modèles qui sont les plus sélectionnés ont plus de chances de se réaliser. Nous sélectionnons pour chaque réalisation x le meilleur modèle par Maximum de Vraisemblance :

$$\forall \quad x, \qquad i(x) = \arg\max_i p(x \mid \mathcal{M}_i) \tag{4.51}$$

$$\forall \quad j, \qquad p(\mathcal{M}_j) = \frac{1}{N_x} \sum_x \delta(j, i(x)) \tag{4.52}$$

$$\delta(j,i) = \begin{cases} 1 & \text{, si } j = i \\ 0 & \text{, sinon} \end{cases} \tag{4.53}$$

Cette estimation de la probabilité des modèles a priori permet ensuite de calculer la probabilité conditionnelle $p(\mathcal{M} \mid X)$ en suivant Eq. 4.50 et la probabilité a priori des données. Nous pourrions réutiliser l'évidence $p(X \mid \mathcal{M})$. Seulement, comme la quantité calculée est proportionnelle à $p(X \mid \mathcal{M})$, nous voulons éviter de la réutiliser. En effet, quand nous calculons l'évidence des modèles, nous calculons une quantité proportionnelle dont le facteur de proportionnalité dépend des réalisations. Cette approximation n'influe pas sur le calcul de l'Eq. 4.50. Néanmoins, quand nous voulons calculer la probabilité a priori $p(X)$, le coefficient de proportionnalité influence beaucoup ce calcul. C'est pourquoi nous

ne faisons pas intervenir l'évidence du modèle dans le calcul de la distribution a priori des données. Par conséquent, nous utilisons une autre méthode d'approximation fondée sur l'estimation d'histogrammes par les fenêtres de Parzen. Pour calculer cette distribution, nous considérons la variable aléatoire $Z = p(\mathcal{M} \mid X)$ et nous estimons $p(Z) = p(X)$. Nous définissons le noyau gaussien sur l'espace $\mathcal{Z}$:

$$\mathcal{K}(z) = \frac{1}{\sqrt{2\pi}} exp(-\frac{z^T z}{2}) \tag{4.54}$$

Alors en notant N_z le nombre de réalisations et Δ une valeur petite par rapport aux valeurs de l'espace $\mathcal{Z}$, nous estimons la probabilité a priori par l'histogramme des fenêtres de Parzen :

$$p(z_0) = \frac{1}{N_z \Delta} \sum_z \mathcal{K}(\frac{z_0 - z}{\Delta}) \tag{4.55}$$

Cette estimation de $p(X)$ ne vérifie pas la relation de Bayes avec la probabilité a priori des modèles $p(\mathcal{M})$. Par la suite, nous laissons tomber cette dernière estimée et nous la considérons seulement comme un appui pour calculer la probabilité d'assignement $p(\mathcal{M} \mid X)$. Enfin, comme nous sommes capables maintenant d'obtenir ces deux probabilités pour n'importe quelle réalisation de champ aléatoire et pour n'importe quel modèle, nous pouvons appliquer l'algorithme de recuit simulé (cf. §4.1.5) pour calculer la courbe débit-distorsion du principe IB. Ensuite, par le critère heuristique de courbure (Eq. 4.41), nous pouvons déterminer le couple débit-distorsion optimal. Il en résulte un clustering qui contient la quantité d'information discriminante suffisante.

4.3.3 Comparaison du recuit simulé et de la croissance exponentielle

Nous revenons brièvement aux conclusions du §4.1.5 sur l'équivalence entre recuit simulé et croissance exponentielle du paramètre de compromis β. Sur des données identiques, nous comparons l'algorithme de recuit simulé et un algorithme semblable fondé sur une croissance exponentielle de β. Nous comparons les courbe débit-distorsion obtenues dans la Figure 4.4. Nous remarquons que les gains sont insignifiants et que les deux procédures donnent approximativement la même courbe débit-distorsion. D'autre part, nous remarquons que la courbe obtenue par le recuit simulé est échantillonnée sur plus de points pour un même résultat. En conséquence, nous préférons par la suite augmenter exponentiellement le paramètre de compromis β pour calculer les fonctions débit-distorsion sans perdre en performance. Cette approximation nous permet d'autre part de gagner en complexité de calculs puisque la courbe est échantillonnée sur moins de points et β_c n'est pas recalculé à chaque fois. Nous montrons, dans la Figure 4.5, l'évolution de la taille effective de l'espace $\tilde{\mathcal{X}}$ en fonction de l'évolution du paramètre de compromis β. La taille de cet espace correspond au nombre de centroïdes identifiables calculés par l'algorithme IB. Nous avons vu que nous pouvions initialiser la taille de cet espace, cependant sa taille effective dépend de β. Tout d'abord, nous remarquons que le nombre effectif de centroïdes augmente quand β augmente. Nous présentons ces évolutions pour les deux algorithmes que nous comparons. Nous voyons que l'algorithme du recuit simulé se stabilise avant chaque transition. Au contraire, dans le cas d'une croissance exponentielle, certaines transitions sont évincées sans changer pour autant les résultats. Pour conclure, nous pourrons approximer l'algorithme du recuit simulé par l'algorithme fondé sur une croissance exponentielle de β.

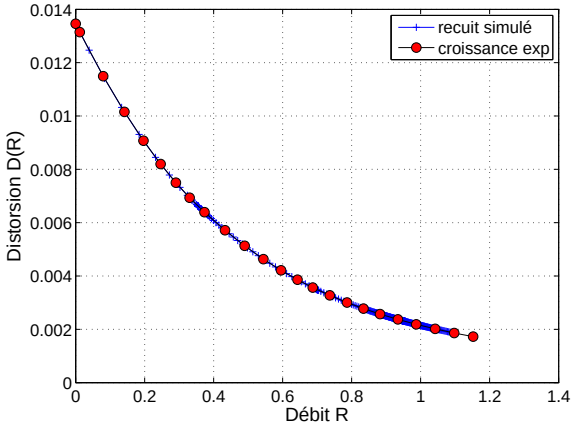


FIG. 4.4 – Les courbes débit-distorsion obtenues avec l'algorithme de recuit simulé et la croissance exponentielle de β sont tracées. Nous remarquons, qu'il y a peu de différences entre ces deux courbes. Par conséquent, la croissance exponentielle nous donne une bonne approximation de la courbe débit-distorsion.

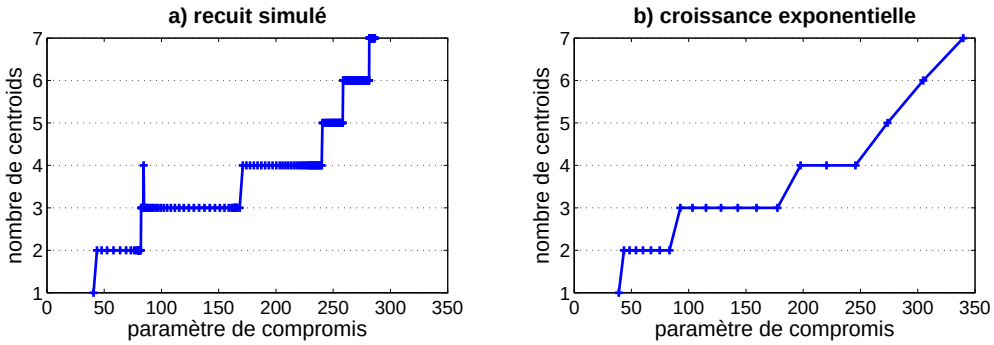


FIG. 4.5 – Nous présentons le nombre effectif de centroïdes en fonction du paramètre de compromis β. Nous constatons que ce nombre croît avec β. (a) Dans le cas du recuit simulé, nous traçons cette courbe. Nous observons que chaque transition est effectuée puisque la courbe se stabilise pour chaque entier. (b) Dans le cas d'une croissance exponentielle de β, nous observons que l'algorithme ne se stabilise pas pour chaque nombre de centroïdes. Les solutions calculées sont toutefois très proches du cas (a).

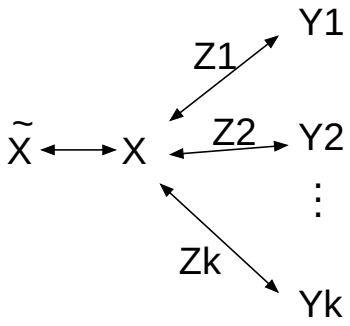


FIG. 4.6 – Ce graphe présente des indépendances markoviennes qui existent entre les variables aléatoires dans le principe de *Multi-Information Bottleneck*. L'information d'importance est composée par plusieurs variables indépendantes. De plus cette information est indépendante de $\tilde{X}$ sachant X.

4.4 Le principe de *Multi-Information Bottleneck* (MIB)

Nous introduisons le principe de *Multi-Information Bottleneck* (MIB) pour extraire plusieurs types d'information. Ce principe généralise le principe IB. Nous présentons les équations consistantes et étendons le critère heuristique fondé sur la courbure. Enfin, nous concluons par décrire deux types d'information d'importance (couleur et évolution des textures) qui peuvent être extraits des STIS. Les expériences réalisées sur les STIS sont décrites au §6.1.4.

4.4.1 Description du principe de *Multi-Information Bottleneck*

Friedman et al. (2001) présentent le principe de *Multivariate Information Bottleneck* qui est dans l'esprit de notre principe de *Multi-Information Bottleneck* (Gueguen & Datcu, 2006). La différence réside dans le fait qu'ils présentent un principe fondé sur les réseaux bayésiens en ne considérant qu'un paramètre de compromis, alors que nous nous appuyons sur plusieurs paramètres de compromis en relation avec un réseau bayésien particulier.
Imaginons que nous avons plusieurs types d'information d'intérêt représentés par plusieurs variables aléatoires $\{Y_i\}_{i\leq N_Y}$. De plus considérons que ces types d'information ne se superposent pas, alors les variables Y_i sont indépendantes. Ainsi, l'information mutuelle partagée par ces variables est nulle. Si ces considérations sont respectées, nous obtenons le graphe des dépendances de Markov (cf. Figure 4.6) entre l'information d'intérêt portée par $\{Y_i\}_{i\leq N_Y}$, nos données X et une représentation compacte $\tilde{X}$. Comme dans le principe IB, nous voulons extraire dans $\tilde{X}$ l'information d'importance contenue dans X, où cette information d'importance est composée par différents types d'information indépendants. Nous transposons le principe d'*Information Bottleneck* (Eq. 4.4) au principe de *Multi-Information Bottleneck* par l'équation suivante :

$$\min_{p(\tilde{x}|x),\tilde{\mathcal{X}}} I(\tilde{X},X) - \sum_{i=1}^{N_Y} \beta_i I(\tilde{X},Y_i) \tag{4.56}$$

Nous introduisons plusieurs paramètres de compromis $\{\beta_i\}$ pour donner de l'importance à tel ou tel type d'information. La mesure de l'information mutuelle $I(\tilde{X},X)$ donne la quantité d'information extraite. D'un autre coté, nous sommes capables de qualifier l'information extraite par les quantités $I(\tilde{X},Y_i)$ et la connaissance de $\{Y_i\}_{i\leq N_Y}$. Nous

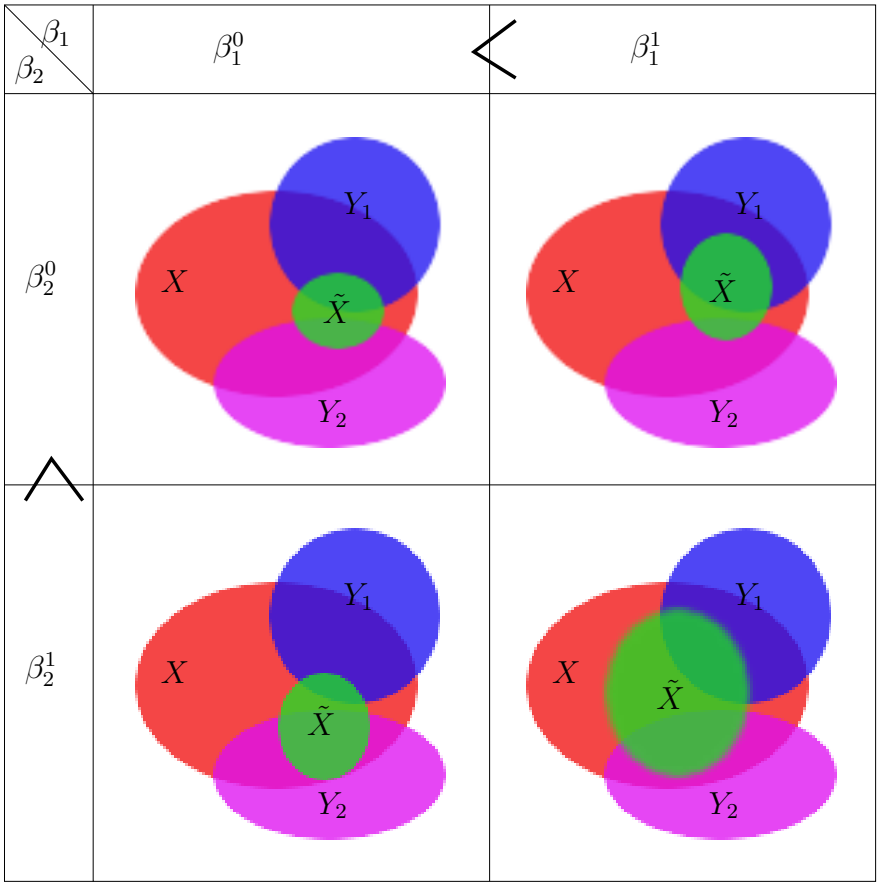


FIG. 4.7 – Nous donnons une représentation abstraite de l'information sous forme d'ensembles. La répartition de l'information est décrite pour le principe de *Multi-Information Bottleneck* avec deux variables d'intérêt Y_1 et Y_2. Nous présentons quatre cas relatifs en fonction des paramètres de compromis β_1 et β_2. Nous avons les relations d'ordre suivantes : $\beta_1^0 < \beta_1^1$ et $\beta_2^0 < \beta_2^1$, qui nous donnent quatre configurations. Quand $\beta_1 = \beta_2$, l'information portée par $\tilde{X}$ dépend de l'amplitude de β_1. Dans le cas où un des paramètres est supérieur à l'autre, c'est l'information liée au plus grand paramètre qui est privilégiée.

présentons un exemple abstrait de la répartition des informations par le principe MIB dans la Figure 4.7. Jusqu'a présent nous avons étudié l'évolution des textures en utilisant le principe IB. Cependant, nous avons bien précisé au §2.2 que plusieurs catégories d'information coexistent dans les images : la couleur, la texture, les formes. Ces types d'information sont considérés comme indépendants et sont traités séparément dans de nombreux systèmes d'extraction d'information. Si tel est le cas, notre principe MIB permet de considérer ces différents types d'information. En conséquence, nous pourrons déterminer quels types d'information prédominent dans notre extraction d'information des STIS.

Le principe MIB engendre un panel de courbes débit-distorsion qui dépendent du type d'information d'importance. Nous donnons les solutions consistantes du problème de *Multi-Information Bottleneck* par la suite pour mettre en avant le partitionnement des informations d'intérêt.

4.4.2 Equations consistantes du principe MIB

Comme dans le cas du principe IB, nous donnons une formulation équivalente du principe MIB en introduisant les variables aléatoires Z_i et $\tilde{Z}_i$ dont les réalisations sont définies par :

$$z_i = p(Y_i \mid x) \tag{4.57}$$
$$\tilde{z}_i = p(Y_i \mid \tilde{x}) \tag{4.58}$$

Comme Z_i et $\tilde{Z}_i$ sont des statistiques suffisantes (Banerjee et al., 2004a), nous avons l'égalité suivante entre les probabilités conditionnelles :

$$\forall i \quad p(\tilde{x} \mid x) = p(\tilde{z}_i \mid z_i) \quad \text{et} \quad p(X) = p(Z_i) \tag{4.59}$$

Pour des raisons pratiques, nous noterons cette probabilité conditionnelle $p(\tilde{x} \mid x) = p(\tilde{z} \mid z)$ qui est indépendante de l'indice i et du type d'information. En considérant ces réécritures des probabilités, le principe MIB se reformule par l'expression suivante :

$$\min_{p(\tilde{z}|z),\tilde{\mathcal{Z}}_i} \sum_{i=1}^{N_Y} \frac{I(Z_i, \tilde{Z}_i)}{N_Y} + \beta_i \mathbb{E}_{Z_i,\tilde{Z}_i} \Big[d_{KL}(Z_i, \tilde{Z}_i) \Big] \tag{4.60}$$

où $\{\tilde{\mathcal{Z}}_i\}$ sont les espaces de reproduction correspondant aux variables aléatoires $\tilde{Z}_i$. Chaque espace de reproduction correspond à la reproduction d'un type d'information. Par des techniques de dérivations du lagrangien, nous sommes capables de calculer les équations consistantes du principe. Nous donnons les deux premières équations consistantes correspondant à la probabilité d'assignement et à la probabilité des centroïdes :

$$p(\tilde{z} \mid z) = \frac{p(\tilde{z})}{N(z, \{\beta_i\})} \exp\left\{ -\sum_{i=1}^{N_Y} \beta_i d_{KL}(z_i, \tilde{z}_i) \right\} \tag{4.61}$$
$$N(z, \{\beta_i\}) = \sum_{\tilde{z}} p(\tilde{z}) \exp\left\{ -\sum_{i=1}^{N_Y} \beta_i d_{KL}(z_i, \tilde{z}_i) \right\} \tag{4.62}$$
$$\tag{4.63}$$

Comme $p(\tilde{z} \mid z)$ et $p(z)$ ne dépendent pas de i, nous sommes sûrs que $p(\tilde{z})$ ne dépend pas du type d'information. Ainsi, nous avons l'équation consistante suivante :

$$p(\tilde{z}) = \sum_{z} p(\tilde{z} \mid z) p(z) \tag{4.64}$$

Nous nous intéressons maintenant à la construction des espaces de reproduction qui dépendent du type d'information dénoté par i. Nous obtenons par conséquent N_Y équations consistantes qui sont exprimées par :

$$\tilde{z}_i = \sum_{z} p(z \mid \tilde{z}) z_i \tag{4.65}$$
$$\tilde{z}_i = \sum_{z_i} p(z_i \mid \tilde{z}_i) z_i \tag{4.66}$$

Si toutes les équations consistantes sont vérifiées, alors nous pouvons écrire les fonctions débit-distorsion suivantes :

$$R(\{\beta_i\}) = \sum_{z,\tilde{z}} p(\tilde{z} \mid z)p(z) \log \frac{p(\tilde{z} \mid z)}{p(\tilde{z})} \tag{4.67}$$

$$D^i(\{\beta_i\}) = \sum_{z_i,\tilde{z}_i} p(\tilde{z}_i \mid z_i)p(z_i)d_{KL}(z_i, \tilde{z}_i) \tag{4.68}$$

$$D^i(\{\beta_i\}) = \sum_{z,\tilde{z}} p(\tilde{z} \mid z)p(z)d_{KL}(z_i, \tilde{z}_i) \tag{4.69}$$

Si N_Y est le nombre de types d'information, alors les $(N_Y + 1)$-uplets $(R, D^1, D^2, .., D^{N_Y})$ forment une hypersurface de l'espace $(\mathbb{R}^+)^{N_Y+1}$. Cette hypersurface permet de séparer l'espace $(\mathbb{R}^+)^{N_Y+1}$ en deux ensembles compactes, dont un ensemble est la région des $(N_Y + 1)$-uplets atteignables par codage. Pour une même quantité d'information notée R, nous pouvons obtenir différentes distorsions. Ainsi, si nous voulons privilégier un type d'information i, il suffit que le paramètre de compromis β_i soit plus grand que les autres paramètres de compromis.
L'algorithme qui découle du principe MIB est fortement similaire à l'algorithme présenté au §4.1.3. C'est un algorithme itératif du type Espérance-Maximisation qui met à jour tout à tour la probabilité d'assignement et les espaces de reproduction. Il suffit de répéter plusieurs fois la construction des espaces de reproduction pour chaque type d'information. Cet algorithme converge vers un minimum local de la fonctionnelle du principe MIB. Nous pourrions adopter une stratégie fondée sur le recuit simulé pour converger vers des minima globaux comme nous l'avons décrit au §4.1.4. Cependant, par manque d'investigation, nous ne savons pas comment se comporte ce type de procédure quand un système possède plusieurs paramètres de température. Aussi, nous préconisons un croissance exponentielle de la norme de $\{\beta_i\}$ selon plusieurs directions de l'espace des paramètres de compromis. Par conséquent, pour une direction donnée du vecteur des paramètres de compromis, le problème MIB rentre dans le cadre décrit par Friedman et al. (2001), et admet un seul paramètre de compromis donné par la norme de $\{\beta_i\}$. Alors, la procédure de recuit simulé peut s'appliquer dans cette configuration.

4.4.3 Quantité d'information d'importance dans le cas MIB

Nous nous plaçons dans le cas du principe de *Multi-Information Bottleneck* normalisé pour ne pas influencer l'importance des informations portées par les variables Y_i. Le critère se réécrit sous la forme suivante :

$$\min_{p(\tilde{x}|x),\tilde{\mathcal{X}}} \frac{I(\tilde{X}, X)}{H(X)} - \sum_{i=1}^{N_Y} \beta_i \frac{I(\tilde{X}, Y_i)}{I(X, Y_i)} \tag{4.70}$$

La première normalisation correspond à diviser l'information mutuelle par l'entropie de X. Alors la quantité $\frac{I(\tilde{X},X)}{H(X)}$ prend ses valeurs dans l'intervalle $[0, 1]$. Quand le débit est maximal, le rapport vaut 1. Les autres normalisations consistent à diviser l'information extraite sur l'information d'importance, $\frac{I(\tilde{X},Y_i)}{I(X,Y_i)}$. Ces quantités prennent aussi leurs valeurs dans l'intervalle $[0, 1]$. C'est équivalent à ce que la distorsion normalisée $d^i = 1 - \frac{I(\tilde{X},Y_i)}{I(X,Y_i)}$ prenne ses valeurs dans $[0, 1]$.

Selon ce critère modifié, si les paramètres de compromis sont égaux, alors les informations véhiculées par chaque variable Y_i auront le même poids lors de la résolution du problème. Pour influer sur l'importance de tel où tel type d'information, nous fixons une direction du vecteur de paramètres de compromis $\vec{\beta} = [\beta_1, ..., \beta_{N_Y}]$. Celui-ci est formé de valeurs positives et il est unitaire, c'est-à-dire que la somme des β_i vaut 1. Seule cette direction permet d'établir les préférences sur les types d'information. Si ce vecteur est fixé, nous pouvons reformuler le principe MIB en fonction d'un seul paramètre de compromis γ :

$$\min_{p(\tilde{x}|x),\tilde{\mathcal{X}}} \frac{I(\tilde{X},X)}{H(X)} - \gamma \sum_{i=1}^{N_Y} \beta_i \frac{I(\tilde{X},Y_i)}{I(X,Y_i)} \tag{4.71}$$

En l'occurrence, nous explorons l'espace des multi-distorsions selon une direction donnée par le vecteur $\vec{\beta}$. Alors, la distorsion globale est donnée par la somme des distorsions normalisées, et le débit normalisé r est donné par la normalisation du débit R. Cela nous donne une courbe débit-distorsion globale et paramétrée pour une direction fixée :

$$d^i(\gamma\vec{\beta}) = \frac{D^i(\gamma\vec{\beta})}{I(X,Y_i)} \tag{4.72}$$

$$d(\gamma) = \sum_{i=1}^{N_Y} \beta_i d^i(\gamma\vec{\beta}) \tag{4.73}$$

$$r(\gamma) = \frac{R(\gamma\vec{\beta})}{H(X)} \tag{4.74}$$

Ces grandeurs sont comparables puisqu'elles prennent leurs valeurs dans l'intervalle $[0,1]$. Par conséquent, nous obtenons une courbe débit-distorsion normalisée dont le point critique peut être calculé en suivant le critère de courbure de l'Eq. 4.41. Ce point critique $\hat{\gamma}$ induit un point critique de l'espace des paramètres qui est donné par le vecteur $\hat{\gamma}\vec{\beta}$.
En résumé, quand les importances des informations sont fixées par $\vec{\beta}$, nous avons un critère heuristique fondé sur la courbure de la fonction débit-distorsion globale pour trouver le vecteur critique $\hat{\gamma}\vec{\beta}$. L'extraction d'information réalisée à ce point critique est optimale pour la pondération $\vec{\beta}$.

4.4.4 Caractérisation de l'évolution de la couleur dans les STIS

Nous avons remarqué dans l'état de l'art (cf. §2.2), que les types d'information extraits dans les images et vidéos sont la couleur, la texture, la forme et le mouvement. En général ces types d'information sont extraits indépendamment les uns des autres, et il est communément admis que ces types d'information sont indépendants. Par exemple, une texture en niveaux de gris peu être teintée en une multitude de couleurs différentes. Il est évident que ces deux types d'information ne partagent aucune information mutuelle. Ainsi dans de nombreux systèmes d'extraction d'information, ces différents types d'information sont fusionnés (cf. §2.5) pour prendre en compte tous les types d'information. Ensuite une réduction d'information permet d'éliminer les redondances. Quand nous admettons que les types d'information sont indépendants, nous ne déduisons pas que les informations sous-jacentes associées à chaque type sont indépendantes. A titre d'exemple, la texture gazon a de forte chance d'être de couleur verte dans les images

naturelles. Cependant, la texture est indépendante de la couleur dans le processus de génération d'une image ou d'une vidéo.
En ce qui concerne les STIS, il n'y pas d'information de mouvement. Ce que nous observons dans la série est plutôt lié aux variations de texture dans le temps. D'ailleurs, c'est cette observation qui nous a amené à considérer les champs aléatoires de Gibbs-Markov et autobinomiaux pour extraire l'information d'importance. Néanmoins, en appliquant le principe d'*Information Bottleneck* à la STIS, nous ne prenons pas en compte l'information de couleur qui est due à l'existence des trois bandes spectrales. Par conséquent, nous voulons extraire conjointement l'information de couleur et texturale des STIS. D'après les suppositions faites précédemment, le principe de *Multi-Information Bottleneck* s'adapte complètement à cette extraction d'information bipartite.
Nous avons déjà donné les modèles d'évolution de textures qui permettent de caractériser les événements spatio-temporels monochromes. Nous exploitons ces modèles pour décrire l'évolution texturale de la bande la plus informative de la séquence. Comme nous le verrons par la suite, la troisième bande spectrale proche infrarouge est la plus informative.
Nous devons maintenant modéliser l'information de couleur dans les séries. Du fait des changements abrupts de réflectances qui existent d'une image à l'autre, nous prenons en compte l'évolution de la couleur au cours du temps. Par cette supposition, nous considérons que l'information de couleur est stationnaire spatialement. Soit X un champ aléatoire de dimensions $l_x \times l_y \times l_t$ où chaque pixel X_s est un vecteur de couleur à trois dimensions. Nous notons chaque image du champ tridimensionnel X_t tel que $t \in [1, ..., l_t]$. Alors nous considérons que les couleurs sont distribuées selon des gaussiennes qui dépendent de chaque instant. Ainsi, nous avons à chaque instant t :

$$p(X_t \mid \mu_t, \sigma_t) \sim \mathcal{N}(\mu_t, \sigma_t) \tag{4.75}$$

où μ_t est le vecteur couleur moyen de l'image X_t et σ_t est la matrice de covariance des couleurs. La probabilité globale du champ aléatoire s'exprime par :

$$p(X \mid \{\mu_t, \sigma_t\}_{t\in[1,...,l_t]}) = \prod_{t\in[1,...,l_t]} p(X_t \mid \mu_t, \sigma_t) \tag{4.76}$$

Alors le signal $\mu = \{\mu_t\}$ représente l'évolution des couleurs au cours du temps. Et la suite de variance $\sigma = \{\sigma_t\}$ représente la dispersion des couleurs au cours du temps. Ainsi, nous pouvons considérer que l'information d'importance est contenue dans le signal de moyenne et de covariance. Goldberger et al. (2002) adoptent le même type de raisonnement pour extraire l'information d'importance des images. Nous pouvons associer à chaque réalisation du champ aléatoire l'estimation des moyennes et des variances $(\hat{\mu}(x), \hat{\sigma}(x))$. La variable aléatoire Y_2 contenant l'information d'importance prend ces valeurs dans l'espace des estimés $\{(\hat{\mu}(x), \hat{\sigma}(x))\}_x$ avec une probabilité équivalente à $p(X)$. Cette supposition permet de calculer les caractéristiques de canal entre X et Y_2 en prenant :

$$p(X \mid y_2) = p(X \mid y_2 = (\hat{\mu}(v), \hat{\sigma}(v))) \tag{4.77}$$

Pour calculer la probabilité conditionnelle $p(Y_2 \mid X)$, nous utilisons les méthodes présentées au §4.3.2.
En résumé, nous avons l'information sur l'évolution texturale qui est représentée par la variable aléatoire Y_1. Cette variable prend ses valeurs dans l'espace des modèles de Gibbs-Markov tels que les champs de Gauss-Markov et autobinomiaux. L'information sur l'évolution des couleurs est portée par la variable Y_2 présentée précédemment. Pour

ne pas influencer sur l'importance des informations, nous appliquons le principe de *Multi-Information Bottleneck* normalisé aux réalisations de X.

4.5 Le principe de *Variational Information Bottleneck* (VIB)

Nous présentons dans cette section, le principe de *Variational Information Bottleneck* (VIB) dérivé du principe IB. C'est un principe dual fondé sur la mesure de la variation de l'information. Nous présentons uniquement le principe théorique qui pourrait être utilisé pour extraire l'information. Tout d'abord, nous présentons la mesure d'information I_α. Ensuite, nous présentons le principe de VIB et les équations consistantes qui en découlent. Enfin, nous concluons sur les intérêts d'un tel principe pour l'extraction d'information.

4.5.1 α-Information

Nous rappelons que la définition de l'information mutuelle est donnée dans l'Eq. 3.68. Meila (2003) donne la définition de la variation de l'information et montre son intérêt pour la comparaison de clusterings. La variation de l'information correspond à mesurer les différences qui existent entre deux variables aléatoires X et Y. La variation de l'information s'exprime alors comme la quantité d'information totale moins l'information mutuelle :

$$\begin{aligned} VI(X,Y) &= H(X,Y) - I(X,Y) \\ &= H(X \mid Y) + H(Y \mid X) \\ &= -\sum_{x,y} p(x,y) \log \frac{p(x,y)^2}{p(x)p(y)} \end{aligned} \tag{4.78}$$

Cette quantité est positive et s'exprime en bits d'information. La variation de l'information est la mesure duale de l'information mutuelle puisqu'elle mesure les différentes informations portées par chaque variable. Nous introduisons une mesure de l'information intermédiaire qui permet de pondérer ces deux quantités d'information. Nous notons cette quantité $I_\alpha(X,Y)$ et donnons sa définition :

$$I_\alpha(X,Y) = I(X,Y) - \alpha H(X,Y) \tag{4.79}$$

Cette quantité peut être positive ou négative. C'est une fonction continue de α. Quand $\alpha = 0$, nous avons $I_\alpha(X,Y) = I(X,Y)$ et cette quantité est positive. Quand $\alpha = 1$, nous avons $I_\alpha(X,Y) = -VI(X,Y)$ et la mesure est négative. Ainsi, cette mesure permet de quantifier l'information mutuelle avec des mesures positives. Elle mesure aussi la variation d'information par des quantités négatives. Par conséquent, en faisant varier continûment le paramètre α de 0 vers 1, I_α mesure des combinaisons d'information mutuelle et de différence. Enfin, il y a un juste milieu entre les différences d'information et l'information mutuelle. Ce juste milieu est donné pour un $\hat{\alpha}$ particulier qui s'exprime par :

$$\hat{\alpha} = \frac{I(X,Y)}{H(X,Y)} \tag{4.80}$$

$$I_{\hat{\alpha}} = 0 \tag{4.81}$$

Cet $\hat{\alpha}$ correspond à avoir autant d'information mutuelle que de variation d'information entre les deux variables. Nous donnons une définition alternative de I_α :

$$I_\alpha(X,Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)^{1+\alpha}}{p(x)p(y)} \tag{4.82}$$

L'α-information moyenne les quantités $\log \frac{p(x,y)^{1+\alpha}}{p(x)p(y)}$ qui peuvent être soit positives soit négatives. Quand cette quantité est positive, (x,y) partagent de l'information. Au contraire, (x,y) contiennent de l'information différente quand cette quantité est négative. En outre dans le cas intermédiaire correspondant à $\hat{\alpha}$, nous pouvons déterminer les couples apportant de l'information commune et les couples apportant de l'information différenciée. En résumé, cette mesure permet d'unifier l'information mutuelle et la variation de l'information.

4.5.2 Le principe d'α-*Information Bottleneck*

Nous généralisons le principe IB avec cette nouvelle mesure de l'information I_α. Nous considérons toujours les mêmes indépendances statistiques entre variables et le fait que Y contienne l'information d'importance. Le principe s'exprime par l'équation suivante :

$$\min_{p(\tilde{x}|x),\tilde{\mathcal{X}}} I_\alpha(\tilde{X},X) - \beta I_\alpha(\tilde{X},Y) \tag{4.83}$$

Ce principe permet alors d'extraire l'information d'importance portée par Y tout en imposant des contraintes sur les différences d'informations. Par exemple, dans le cas où $\alpha = 1$, nous obtenons le principe de *Variational Information Bottleneck* exprimé par :

$$\max_{p(\tilde{x}|x),\tilde{\mathcal{X}}} VI(\tilde{X},X) - \beta VI(\tilde{X},Y) \tag{4.84}$$

Ce dernier principe revient à construire $\tilde{X}$ tel que celui-ci soit le plus différent de X tout en ayant le moins de différence possible avec Y. Ainsi, par ce principe nous ciblons exactement l'information partagée par X et Y. Par exemple, quand β est grand, $\tilde{X}$ a tendance à recouvrir Y. Par conséquent, l'information de $\tilde{X}$ varie par rapport à X contrairement au cas de l'information mutuelle. Dans le cas général où $\alpha \in]0,1[$, le principe est se situe entre les deux principes extrêmes IB et VIB. Il est plus difficile d'en donner une interprétation. Comme pour le principe d'*Information Bottleneck*, nous réécrivons le problème sous la forme suivante par la transformation de l'espace de reproduction :

$$\min_{p(\tilde{x}|x),p(y|\tilde{x})} I_\alpha(\tilde{X},X) - \beta I_\alpha(\tilde{X},Y) \tag{4.85}$$

Ce type de problème ne rentre pas dans le cadre de la théorie débit-distorsion et ne permet pas d'appliquer les équations consistantes décrites dans le chapitre précédent. Nous opérons donc au calcul des équations consistantes par minimisation directe du lagrangien.

4.5.3 Equations consistantes (première étape)

Nous faisons la minimisation séparément sur $p(\tilde{x} \mid x)$ et $p(y \mid \tilde{x})$. Nous considérons que la chaîne de Markov $\widetilde{X} \leftrightarrow X \leftrightarrow Y$ est vérifiée. Par conséquent, $\widetilde{X}$ et Y sont indépen-

dants connaissant X et nous avons les équations suivantes.

$$p(\tilde{x} \mid y) = \sum_x p(x \mid y)p(\tilde{x} \mid x) \tag{4.86}$$

$$p(\tilde{x}) = \sum_x p(\tilde{x} \mid x)p(x) \tag{4.87}$$

Les équations précédentes induisent les dérivées suivantes selon $p(\tilde{x} \mid x)$:

$$\frac{\partial p(\tilde{x} \mid y)}{\partial p(\tilde{x} \mid x)} = p(x \mid y) \tag{4.88}$$

$$\frac{\partial p(\tilde{x})}{\partial p(\tilde{x} \mid x)} = p(x) \tag{4.89}$$

En introduisant, les multiplicateurs de Lagrange $\lambda(x)$ pour la normalisation des distributions conditionnelles $p(\tilde{x} \mid x)$, la fonctionnelle à minimiser devient :

$$\mathcal{L} = I_\alpha(X, \widetilde{X}) - \beta I_\alpha(\widetilde{X}, Y) - \sum_{x,\tilde{x}} \lambda(x)p(\tilde{x} \mid x) \tag{4.90}$$

Nous dérivons cette fonction par rapport à la quantité $p(\tilde{x} \mid x)$ pour x et $\tilde{x}$ donnés. Pour une meilleure visibilité, nous donnons la dérivée des trois termes séparément. Tout d'abord, nous donnons la dérivée de $I_\alpha(X, \widetilde{X})$:

$$I_\alpha(X, \widetilde{X}) = \sum_{x,\tilde{x}} p(\tilde{x} \mid x)p(x) \log p(\tilde{x} \mid x)^{1+\alpha} p(x)^\alpha - \sum_{\tilde{x}} p(\tilde{x}) \log p(\tilde{x}) \tag{4.91}$$

$$\begin{aligned} \frac{\partial I_\alpha(X, \widetilde{X})}{\partial p(\tilde{x} \mid x)} &= p(x)[1 + \alpha + \log p(\tilde{x} \mid x)^{1+\alpha} p(x)^\alpha] \\ &\quad - \frac{\partial p(\tilde{x})}{\partial p(\tilde{x} \mid x)}[1 + \log p(\tilde{x})] \end{aligned} \tag{4.92}$$

$$\frac{\partial I_\alpha(X, \widetilde{X})}{\partial p(\tilde{x} \mid x)} = p(x)[\log \frac{p(\tilde{x} \mid x)^{1+\alpha} p(x)^\alpha}{p(\tilde{x})} + \alpha] \tag{4.93}$$

Deuxièmement, nous donnons la dérivée de $I_\alpha(\widetilde{X}, Y)$:

$$I_\alpha(\widetilde{X}, Y) = \sum_{y,\tilde{x}} p(\tilde{x} \mid y)p(y) \log p(\tilde{x} \mid y)^{1+\alpha} p(y)^\alpha - \sum_{\tilde{x}} p(\tilde{x}) \log p(\tilde{x}) \tag{4.94}$$

$$\begin{aligned} \frac{\partial I_\alpha(\widetilde{X}, Y)}{\partial p(\tilde{x} \mid x)} &= \sum_y p(y) \frac{\partial p(\tilde{x} \mid y)}{\partial p(\tilde{x} \mid x)}[1 + \alpha + \log p(\tilde{x} \mid y)^{1+\alpha} p(y)^\alpha] \\ &\quad - \frac{\partial p(\tilde{x})}{\partial p(\tilde{x} \mid x)}[1 + \log p(\tilde{x})] \end{aligned} \tag{4.95}$$

$$\frac{\partial I_\alpha(\widetilde{X}, Y)}{\partial p(\tilde{x} \mid x)} = p(x)[\sum_y p(y \mid x) \log p(\tilde{x} \mid y)^{1+\alpha} p(y)^\alpha - \log p(\tilde{x}) + \alpha] \tag{4.96}$$

$$\frac{\partial I_\alpha(\widetilde{X}, Y)}{\partial p(\tilde{x} \mid x)} = p(x)[\sum_y p(y \mid x) \log \frac{p(y \mid \tilde{x})^{1+\alpha}}{p(y)} + \log p(\tilde{x})^\alpha + \alpha] \tag{4.97}$$

Finalement, la dérivée du lagrangien $\frac{\partial \mathcal{L}}{\partial p(\tilde{x}|x)}$ est donnée par :

$$p(x)\left\{\log\frac{p(\tilde{x}\mid x)^{1+\alpha}p(x)^{\alpha}}{p(\tilde{x})}+\alpha-\beta[\sum_{y}p(y\mid x)\log\frac{p(y\mid\tilde{x})^{1+\alpha}}{p(y)}+\log p(\tilde{x})^{\alpha}+\alpha]-\frac{\lambda(x)}{p(x)}\right\} \tag{4.98}$$

Cette quantité se réécrit :

$$p(x)\left\{\log\frac{p(\tilde{x}\mid x)^{1+\alpha}p(x)^{\alpha}}{p(\tilde{x})^{1+\beta\alpha}}+(1-\beta)\alpha-\beta\sum_{y}p(y\mid x)\log\frac{p(y\mid\tilde{x})^{1+\alpha}}{p(y)}-\frac{\lambda(x)}{p(x)}\right\} \tag{4.99}$$

Nous pouvons introduire de nouveaux multiplicateurs de Lagrange avec tout les termes indépendants de $\tilde{x}$ et qui dépendent uniquement de x. Nous notons que $I_\alpha(x,Y) = \sum_y p(y\mid x)\log\frac{p(y|x)^{1+\alpha}p(x)^{\alpha}}{p(y)}$ dépend seulement de x et peut être absorbé dans le nouveau multiplicateur :

$$\tilde{\lambda}(x)=\frac{\lambda(x)}{p(x)}-\log p(x)^{\alpha}-(1-\beta)\alpha-\beta I_{\alpha}(x,Y) \tag{4.100}$$

La dérivée du lagrangien peut alors se réécrire par :

$$\frac{\partial\mathcal{L}}{\partial p(\tilde{x}\mid x)}=p(x)\left\{\log\frac{p(\tilde{x}\mid x)^{1+\alpha}}{p(\tilde{x})^{1+\beta\alpha}}+\beta(1+\alpha)\sum_{y}p(y\mid x)\log\frac{p(y\mid x)}{p(y\mid\tilde{x})}-\tilde{\lambda}(x)\right\}=0 \tag{4.101}$$

La résolution de la dernière équation nous donne la première équation consistante :

$$p(\tilde{x}\mid x)=\frac{p(\tilde{x})^{\frac{1+\alpha\beta}{1+\alpha}}}{N_{\alpha}(x,\beta)}\exp\left\{-\beta\sum_{y}p(y\mid x)\log\frac{p(y\mid x)}{p(y\mid\tilde{x})}\right\} \tag{4.102}$$

où la fonction de normalisation est donnée par :

$$N_{\alpha}(x,\beta)=\sum_{\tilde{x}}p(\tilde{x})^{\frac{1+\alpha\beta}{1+\alpha}}\exp\left\{-\beta\sum_{y}p(y\mid x)\log\frac{p(y\mid x)}{p(y\mid\tilde{x})}\right\} \tag{4.103}$$

D'autre part, $p(\widetilde{X})$ dépend de $p(\widetilde{X}\mid X)$ par la relation de Bayes, et donne la seconde équation consistante (cf. Eq. 4.87)

4.5.4 Equations consistantes (seconde étape)

Nous supposons que Eq. 4.87 est vérifiée et que $p(\widetilde{X})$ est fixée. Nous dérivons maintenant le lagrangien par rapport à la distribution conditionnelle $p(y\mid\tilde{x})$. Nous substituons Eq. 4.102 dans Eq. 4.83 et nous obtenons la fonctionnelle suivante :

$$\mathcal{L}=-\sum_{x}p(x)\log N_{\alpha}(x,\beta)-\sum_{\tilde{x},y}\lambda(\tilde{x})p(y\mid\tilde{x}) \tag{4.104}$$

Nous dérivons cette fonction par rapport à $p(y\mid\tilde{x})$ pour y et $\tilde{x}$ donnés :

$$\frac{\partial\mathcal{L}}{\partial p(y\mid\tilde{x})}=-\sum_{x}p(x)\frac{\frac{\partial N_{\alpha}(x,\beta)}{\partial p(y|\tilde{x})}}{N_{\alpha}(x,\beta)}-\lambda(\tilde{x}) \tag{4.105}$$

Dans un premier temps, nous donnons la dérivée du terme de normalisation $N_\alpha(x, \beta)$:

$$\frac{\partial N_\alpha(x,\beta)}{\partial p(y \mid \tilde{x})} = -\beta p(\tilde{x})^\xi \frac{\partial d_{KL}(x,\tilde{x})}{\partial p(y \mid \tilde{x})} \exp\left[-\beta d_{KL}(x,\tilde{x})\right] \tag{4.106}$$

où ξ est exprimé par :

$$\xi = \frac{1+\alpha\beta}{1+\alpha} \tag{4.107}$$

$$d_{KL}(x,\tilde{x}) = \sum_y p(y \mid x) \log \frac{p(y \mid x)}{p(y \mid \tilde{x})} \tag{4.108}$$

$$\frac{\partial d_{KL}(x,\tilde{x})}{\partial p(y \mid \tilde{x})} = \frac{p(y \mid x)}{p(y \mid \tilde{x})} \tag{4.109}$$

Par conséquent, nous pouvons réécrire le lagrangien avec les équations précédentes :

$$\frac{\partial \mathcal{L}}{\partial p(y \mid \tilde{x})} = \beta \sum_x p(x) p(\tilde{x} \mid x) \frac{p(y \mid x)}{p(y \mid \tilde{x})} - \lambda(\tilde{x}) = 0 \tag{4.110}$$

Nous obtenons la distribution $p(y \mid \tilde{x})$ après la normalisation des multiplicateurs de Lagrange. Cette équation est la troisième équation consistante :

$$p(y \mid \tilde{x}) = \sum_x p(x \mid \tilde{x}) p(y \mid x) \tag{4.111}$$

En conclusion, les équations consistantes obtenues par la minimisation du principe α-*Information Bottleneck* sont données dans Eq. 4.102, Eq. 4.87 et Eq. 4.111. En comparaison avec les solutions du principe IB, seule la probabilité conditionnelle $p(\widetilde{X} \mid X)$ admet une définition différente. En effet, celle-ci dépend de $p(\widetilde{X})^\xi$. Pour $\alpha = 0$, nous retombons sur les équations consistantes du problème d'*Information Bottleneck* puisque $\xi = 1$. D'autre part, quand $\alpha = 1$, nous avons les solutions consistantes du principe VIB (cf. Eq. 4.84) et nous avons $\xi = \frac{1+\beta}{2}$. Quand β augmente, la fonction $p(\widetilde{X})^\xi$ a tendance à s'aplanir. C'est comme si nous rendions artificiellement uniforme la distribution de $\widetilde{X}$. Alors, la distribution conditionnelle d'assignement est plus influencée par la divergence de Kullback-Leibler.

4.5.5 Le principe VIB pour l'extraction d'information

Le principe VIB est le problème dual du principe IB. Considérons une variable Y qui véhicule une information de désintérêt. C'est-à-dire que nous ne trouvons aucune importance à cette information. Alors, nous voulons extraire de X une information portée par $\tilde{X}$ qui ne soit pas sans importance. Cette extraction d'information est formalisée par le principe VIB suivant :

$$\max_{p(\tilde{x}|y), p(x|\tilde{x})} VI(\tilde{X}, Y) - \beta VI(\tilde{X}, X) \tag{4.112}$$

où la distribution $p(X \mid Y)$ est donnée a priori. Ainsi, nous voulons maximiser les différences entre $\widetilde{X}$ et Y tout en restreignant ces différences entre $\widetilde{X}$ et X.
Aborder l'extraction d'information comme extraire une partie de l'information sachant

ce qui n'est pas important est une approche très rare. Cette idée est difficile à conceptualiser, mais pourrait être appliquée dans le domaine de l'extraction d'information par l'identification de l'information de désintérêt. Cette stratégie d'exploration de l'information en fonction de ce qu'on ne souhaite pas, pourrait apporter un moyen de découvrir de nouvelles connaissances dans les données.
Les résultats de la minimisation $p(x \mid \tilde{x})$ permet de définir un clustering par Maximum de Vraisemblance. De plus, la connaissance des deux distributions conditionnelles permet d'établir un canal de communication de l'information de désintérêt Y vers les données X.

4.6 Résumé

Dans un premier temps, nous avons présenté le principe d'*Information Bottleneck* pour l'extraction d'information d'une variable aléatoire. Les algorithmes permettant d'extraire l'information ont été décrits en s'appuyant sur les équations consistantes. De plus, nous avons introduit la procédure de recuit simulé pour converger vers des minima globaux de la courbe débit-distorsion. Ensuite, nous nous sommes attachés à décrire l'information d'importance portée par les modèles. Dans ce cas, l'algorithme d'InformationBottleneck intègre une sélection floue de modèle. D'autre part, nous avons décrit un critère heuristique de sélection de couple (R, D) dont l'efficacité a été expérimentalement validé sur des données synthétiques. Puisque ce critère permet de sélectionner un couple (R, D) optimal, il détermine le clustering flou optimal et de ce fait un nombre optimal de clusters. Enfin, pour qualifier l'information d'importance, nous exploitaons les champs aléatoires de Gauss-Markov et autobinomiaux. Ainsi, nous avons voulu faire ressortir l'information relative à l'évolution des textures, contenue dans les champs spatio-temporels.
Dans un second temps, nous avons présenté le principe de *Multi-Information Bottleneck*. Ce principe est une extension du principe d'*Information Bottleneck* et il permet de prendre en compte plusieurs types d'information indépendants. De plus, il permet de privilégier un type d'information lors de l'extraction d'information. Pour ce principe, nous avons étendu le critère de courbure pour déterminer la quantité d'information optimale. Finalement, nous décrivons comment appliquer la méthodologie liée à ce principe sur les champs spatio-temporels en considérant deux types d'information. En effet, nous voulons extraire conjointement les informations de couleur et d'évolution texturale.
Dans un dernier temps, nous décrivons un méta principe fondé sur l'α-information. Cette mesure permet d'unifier l'information mutuelle et la variation de l'information. Nous donnons les équations consistantes qui sont obtenues à partir de ce principe et remarquons qu'elles sont fortement semblables aux équations du principe IB. Ce méta principe nous permet d'obtenir le principe de *Variational Information Bottleneck* qui est le dual du problème originel. Enfin, nous montrons que ce principe peut être utilisé pour l'extraction d'information en mettant de coté l'information non intéressante ou non pertinente connue a priori.
En conclusion, nous avons présenté deux premières méthodologies fondées sur les principes IB et MIB qui permettent d'extraire des STIS une quantité d'information suffisante et d'intérêt pour caractériser et indexer les événements spatio-temporels.

Chapitre 5

Extraction d'information fondée sur les complexités

Dans ce chapitre, nous présentons une nouvelle méthodologie pour extraire l'information fondée sur la complexités de Kolmogorov. Cette méthode permet de produire un code d'une base d'objets qui inclut un index du contenu informationnel.
Dans un premier temps, nous présentons un distance informationnelle inspirée de l'Eq. 3.129 en y intégrant la décomposition LDM. Ensuite, nous présentons un critère qui permet de coder une base d'objets en exploitant les redondances. L'algorithme sous-optimal qui en découle tire parti d'un codeur sans perte et de la distance précédemment introduite pour incorporer l'index au code. Enfin, dans le but d'appliquer cette méthode à des signaux trimensionnels, nous décrivons un codeur sans perte qui donne une décomposition en deux parties de l'information en suivant le principe LDM.

5.1 Intégration du principe LDM dans la mesure de similarité informationnelle

5.1.1 Nouvelle similarité fondée sur les modèles

Nous rappelons que le principe LDM s'énonce dans la théorie de Kolmogorov comme la description en deux parties de longueur minimale (cf. §3.5.3). Nous faisons la supposition que pour chaque objet informatif x, il existe une description en deux parties ou un modèle $\mathcal{M}_x$ tel que l'égalité suivante soit respectée :

$$K(x) \stackrel{+}{=} K(x \mid \mathcal{M}_x) + K(\mathcal{M}_x) \tag{5.1}$$

Dans un cadre statistique, le modèle $\mathcal{M}_x$ s'interprète comme une statistique suffisante (Grünwald & Vitanyi, 2004), et elle se suffit à elle-même pour décrire les propriétés de x. Pour un couple d'objets (x, y), nous avons un modèle $\mathcal{M}_{x,y}$ qui vérifie l'Eq. 5.1. Ces suppositions permettent de donner une nouvelle formulation de la distance informationnelle $d(x, y)$ (cf. Eq. 3.129) en remplaçant les mesures de complexité par les longueurs de codes en deux parties. Nous obtenons, comme première expression de la distance, la formule suivante :

$$\frac{K(x, y \mid \mathcal{M}_{x,y}) + K(\mathcal{M}_{x,y}) - \min\{K(x \mid \mathcal{M}_x) + K(\mathcal{M}_x), K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)\}}{\max\{K(x \mid \mathcal{M}_x) + K(\mathcal{M}_x), K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)\}} \tag{5.2}$$

Pour simplifier l'expression de la distance précédente, nous considérons le cas où $K(x) \leq K(y)$. Le cas opposé est obtenu facilement en inversant x et y. L'expression de la distance devient :

$$d(x,y) = \frac{K(x,y \mid \mathcal{M}_{x,y}) + K(\mathcal{M}_{x,y}) - K(x \mid \mathcal{M}_x) - K(\mathcal{M}_x)}{K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)} \tag{5.3}$$

$$\begin{aligned} d(x,y) &= \frac{K(x,y \mid \mathcal{M}_{x,y}) - K(x \mid \mathcal{M}_x)}{K(y \mid \mathcal{M}_y)} \frac{K(y \mid \mathcal{M}_y)}{K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)} \\ &+ \frac{K(\mathcal{M}_{x,y}) - K(\mathcal{M}_x)}{K(\mathcal{M}_y)} \frac{K(\mathcal{M}_y)}{K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)} \end{aligned} \tag{5.4}$$

Nous remarquons que :

$$\frac{K(y \mid \mathcal{M}_y)}{K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)} + \frac{K(\mathcal{M}_y)}{K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)} = 1 \tag{5.5}$$

Nous introduisons une nouvelle variable α définie par :

$$\alpha = \frac{K(y \mid \mathcal{M}_y)}{K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)} = \frac{K(y \mid \mathcal{M}_y)}{K(y)} \tag{5.6}$$

où α appartient à l'intervalle $[0,1]$. Nous réécrivons l'Eq. 5.4 comme :

$$d(x,y) = \alpha \frac{K(x,y \mid \mathcal{M}_{x,y}) - K(x \mid \mathcal{M}_x)}{K(y \mid \mathcal{M}_y)} + (1-\alpha) \frac{K(\mathcal{M}_{x,y}) - K(\mathcal{M}_x)}{K(\mathcal{M}_y)} \tag{5.7}$$

Nous notons que la distance de similarité est divisée en deux parties. La partie gauche mesure la distance entre objets exprimés dans leur modèle, tandis que la partie droite mesure la similarité entre modèles. Quand α tend vers 0, la mesure se concentre vers les modèles puisque toute l'information est contenue dans le modèle. Dans le cas opposé, le raisonnement contraire convient aussi.
A partir de l'égalité précédente, nous définissons dans le cas général la mesure de similarité suivante (Gueguen & Datcu, 2007b) :

$$\delta(x,y) = \alpha \frac{K(x,y \mid \mathcal{M}_{x,y}) - K(x \mid \mathcal{M}_x)}{K(y \mid \mathcal{M}_y)} + (1-\alpha) \frac{K(\mathcal{M}_{x,y}) - K(\mathcal{M}_x)}{K(\mathcal{M}_y)} \tag{5.8}$$

où $K(x \mid \mathcal{M}_x) + K(\mathcal{M}_x) \leq K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)$ et $\alpha = \frac{K(y|\mathcal{M}_y)}{K(y)}$. Dans l'autre cas, où $K(x \mid \mathcal{M}_x) + K(\mathcal{M}_x) \geq K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)$, nous avons $\alpha = \frac{K(x|\mathcal{M}_x)}{K(x)}$ et l'expression de la mesure de similarité devient :

$$\delta(x,y) = \alpha \frac{K(x,y \mid \mathcal{M}_{x,y}) - K(y \mid \mathcal{M}_y)}{K(x \mid \mathcal{M}_x)} + (1-\alpha) \frac{K(\mathcal{M}_{x,y}) - K(\mathcal{M}_y)}{K(\mathcal{M}_x)} \tag{5.9}$$

Pour définir cette mesure, nous ne supposons pas que l'Eq. 5.1 soit vérifiée. Nous élargissons les contraintes d'égalités qui se traduisent par :

$$\forall x \quad \mid K(x \mid \mathcal{M}_x) + K(\mathcal{M}_x) - K(x) \mid \leq c \tag{5.10}$$

Ces contraintes imposent que la représentation en deux parties soit obtenue à une constante près ne dépendant pas des objets. Par conséquent, nous obtenons les bornes suivantes pour la mesure de similarité en deux parties :

$$\mid \delta(x,y) - \frac{1}{1-\frac{c}{\eta}} d(x,y) \mid \leq \frac{c}{1-\frac{c}{\eta}} \tag{5.11}$$

où $\eta = \max\{K(x \mid \mathcal{M}_x) + K(\mathcal{M}_x), K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)\}$. Le coefficient $\frac{1}{1-\frac{c}{\eta}}$ est supérieur à 1 quand $c \leq \eta$. Cela implique que la mesure $\delta(x, y)$ a de fortes chances d'être supérieure à $d(x, y)$ tout en lui étant proportionnelle. La cas favorable est obtenu quand $c << \eta$, puisque nous obtenons l'égalité :

$$\delta(x, y) \stackrel{+}{=} d(x, y) + O(1) \tag{5.12}$$

En l'occurrence, la mesure $\delta(x, y)$ se rapproche de la distance universelle quand le principe LDM est vérifié. Si nous appliquons le principe LDM pour inférer les modèles de chaque objet, nous nous attendons à obtenir une bonne approximation de $d(x, y)$, puisque nous approchons des conditions favorables.

5.1.2 L'information d'importance

Nous avons spécifié au §3.5.3 et §4.2.1 que l'information pertinente est contenue dans le modèle. L'autre partie correspond au caractère aléatoire de l'objet dans le modèle spécifié. Par conséquent, les deux parties de la décomposition LDM d'un objet x ne véhiculent pas le même type d'information par rapport à un troisième objet y. De ce fait, nous supposons que la partie aléatoire $x \mid \mathcal{M}_x$ ne contribue pas à apporter de l'information à y puisque cette information est de nature aléatoire. Cette supposition entraîne une simplification de la complexité conditionnelle des deux objets x, y :

$$K(y \mid x) = K(y \mid \{x \mid \mathcal{M}_x, \mathcal{M}_x\}) = K(y \mid \mathcal{M}_x) \tag{5.13}$$

Du fait de cette simplification, la mesure de similarité se réécrit comme :

$$d(x, y) = \frac{\max\{K(y \mid \mathcal{M}_x), K(x \mid \mathcal{M}_y)\}}{\max\{K(y), K(x)\}} \tag{5.14}$$

Dans le cas de la similarité en deux parties, l'approximation de l'Eq. 5.13 permet d'approximer les complexités conjointes et les modèles conjoints. Nous considérons les deux objets $z = y \mid M_x$ et $w = x \mid M_y$. Si z (respectivement w) est de nature aléatoire, alors $\mathcal{M}_x$ (respectivement $\mathcal{M}_y$) représente d'autant mieux les propriétés de y (respectivement x). Nous admettons que les deux objets z et w peuvent être décomposés en deux parties pour s'approcher au mieux de l'égalité LDM. Par conséquent dans le cas où $K(x \mid \mathcal{M}_x) + K(\mathcal{M}_x) \leq K(y \mid \mathcal{M}_y) + K(\mathcal{M}_y)$, la mesure de similarité devient :

$$\delta(x, y) = \alpha \frac{K(z \mid \mathcal{M}_z)}{K(y \mid \mathcal{M}_y)} + (1 - \alpha) \frac{K(\mathcal{M}_z)}{K(\mathcal{M}_y)} \tag{5.15}$$

En résumé, l'approximation, selon laquelle la partie aléatoire de l'information n'a pas d'importance, permet de définir les deux mesures de similarités précédentes fondées sur la décomposition de codage en deux parties.

5.1.3 Application de la mesure en deux parties aux modèles stochastiques

Nous nous plaçons dans le cadre du principe LDM énoncé par Rissanen (1983) où des modèles paramétriques sont considérés. Nous rappelons que l'inférence en deux niveaux constitue un codage en deux parties optimal au sens LDM. Soit $\{\mathcal{M}_i\}$ l'ensemble des modèles paramétriques dont les espaces de paramètres sont de dimensions respectives

$\{k_i \in \mathbb{N}^+\}$. Pour la réalisation d'un processus stochastique x^n de longueur n, l'estimateur LDM est donné par :

$$(\hat{i}(x^n), \hat{\theta}(x^n)) = \arg\min_{i,\theta} \{-\log p(x^n \mid \theta, \mathcal{M}_i) + \frac{k}{2} \log \frac{n}{2\pi}\} \tag{5.16}$$

Considérons une seconde réalisation du processus, notée y^m dont la taille est m. Nous admettons pouvoir calculer la complexité conditionnelle suivante pour n'importe quelle réalisation y^m :

$$-\log p(y^m \mid \hat{\theta}(x^n), \mathcal{M}_{\hat{i}(x^n)}) \tag{5.17}$$

Dans le cas d'une prédiction (cf. §3.1.3), cela correspond à faire intervenir un signal d'erreur $z^m = y^m \mid \hat{\theta}(x^n), \mathcal{M}_{\hat{i}(x^n)}$. Dans le cas d'une estimation de paramètres, ce signal d'erreur est supposé être i.i.d. Or ce n'est pas le cas ici, puisque ce n'est pas une estimation. Nous admettons que z^m contient les propriétés de y^m après avoir fait disparaître les propriétés de x^n. Le signal z^m admet une décomposition en deux parties d'après le principe LDM. Ainsi dans le cas où la longueur de code en deux parties de x^n est plus petite que celle de y^m, la mesure de similarité $\delta(x^n, y^m)$ s'exprime par :

$$\delta(x^n, y^m) = \alpha \frac{-\log p(z^m \mid \hat{\theta}(z^m), \mathcal{M}_{\hat{i}(z^m)})}{-\log p(y^m \mid \hat{\theta}(y^m), \mathcal{M}_{\hat{i}(y^m)})} + (1-\alpha) \frac{k_{\hat{i}(z^m)}}{k_{\hat{i}(y^m)}} \tag{5.18}$$

$$\alpha = \frac{-\log p(y^m \mid \hat{\theta}(y^m), \mathcal{M}_{\hat{i}(y^m)})}{-\log p(y^m \mid \hat{\theta}(y^m), \mathcal{M}_{\hat{i}(y^m)}) + \frac{k_{\hat{i}(y^m)}}{2} \log \frac{m}{2\pi}} \tag{5.19}$$

Dans le cas contraire, où le code en deux parties de x^n est plus grand que celui de y^m, nous introduisons $w^n = x^n \mid \hat{\theta}(y^m), \mathcal{M}_{\hat{i}(y^m)}$ pour exprimer la similarité :

$$\delta(x^n, y^m) = \alpha \frac{-\log p(w^n \mid \hat{\theta}(w^n), \mathcal{M}_{\hat{i}(w^n)})}{-\log p(x^n \mid \hat{\theta}(x^n), \mathcal{M}_{\hat{i}(x^n)})} + (1-\alpha) \frac{k_{\hat{i}(w^n)}}{k_{\hat{i}(x^n)}} \tag{5.20}$$

$$\alpha = \frac{-\log p(x^n \mid \hat{\theta}(x^n), \mathcal{M}_{\hat{i}(x^n)})}{-\log p(x^n \mid \hat{\theta}(x^n), \mathcal{M}_{\hat{i}(x^n)}) + \frac{k_{\hat{i}(x^n)}}{2} \log \frac{n}{2\pi}} \tag{5.21}$$

Dans le cadre de l'extraction d'information, nous avons résumé les mesures de similarité les plus employées (cf. §2.5.3). La distance euclidienne impose que les espaces de paramètres soient identiques, c'est-à-dire que les objets soient comparés à partir d'un unique modèle. La distance de Kullback-Leibler compare les paramètres sur un ensemble de données tests identiques. Cependant, cette distance permet d'avoir des espaces de paramètres différents. Enfin, la mesure que nous avons introduite, permet de comparer deux processus stochastiques au travers de plusieurs modèles tout en admettant des espaces de paramètres différents.
Nous rappelons que l'estimation LDM peut être vue comme une compression avec pertes dans le cadre des complexités de Kolmogorov (cf §3.5.4). Ainsi, l'information extraite est portée par les modèles et les paramètres extraits lors de l'estimation en deux niveaux d'inférence. En conséquence, notre distance permet de comparer des objets en exploitant l'information d'importance extraite au sens LDM.

5.1.4 Fouille dans les objets compressés en deux parties

Dans le cas, où il existe un codeur universel en deux parties pour n'importe quel objet informatif x, nous pouvons approximer les complexités de Kolmogorov par les longueurs de code, telles que $C_1(x \mid \mathcal{M})$ est la première longueur de code nécessaire pour représenter x sachant $\mathcal{M}$ et $C_2(\mathcal{M})$ est la longueur de code du modèle. Si un ensemble d'objets est compressé en codant chaque objet optimalement en deux parties, il est possible de retrouver les objets similaires à un exemple sans tout décoder. Il suffit de calculer la similarité δ entre l'exemple et chaque objet de la base. Nous appelons cette distance la Distance Normalisée Suffisante de Compression (DNSC) :

$$DNSC(x,y) = \frac{\max\{C_1(x \mid \mathcal{M}_y), C_1(y \mid \mathcal{M}_x)\}}{\max\{C_1(x \mid \mathcal{M}_x) + C_2(\mathcal{M}_x), C_1(y \mid \mathcal{M}_y) + C_2(\mathcal{M}_y)\}} \tag{5.22}$$

Cette mesure est une version simplifiée, où les signaux intermédiaires z et w ne sont pas pris en compte.
Il y a deux cas. Si l'exemple est plus complexe que l'objet de l'ensemble, il n'est pas nécessaire de décoder l'objet complètement pour calculer la similarité. Seul le modèle doit être décodé. Au contraire si l'objet est plus complexe que l'exemple, il faut décoder celui-ci complètement avant de calculer la distance. Pour éviter, ce second cas de figure, il est possible de considérer des exemples plus complexes que les objets ou de considérer uniquement l'information extraite. Dans ce dernier cas, la mesure de similarité se calcule entre un modèle $\mathcal{M}_x$ et l'exemple y :

$$DNSC(\mathcal{M}_x, y) = \frac{C_1(y \mid \mathcal{M}_x)}{C_1(y \mid \mathcal{M}_y) + C_2(\mathcal{M}_y)} \tag{5.23}$$

La fouille consiste alors à retrouver les propriétés extraites et contenues dans les modèles qui correspondent le plus à l'exemple. Ensuite, il devient évident de prendre l'objet, qui a engendré le modèle, comme objet similaire à l'exemple puisqu'il partage les mêmes propriétés.
En résumé, nous avons décrit une mesure de similarité qui permet d'approximer la similarité de Li et al. (2004) dans le cas des modèles stochastiques ou des codeurs universels en deux parties. Cette similarité possède comme avantages d'exploiter l'information pertinente contenue dans les objets et de permettre une fouille efficace sur des objets compressés.

5.2 Extraction d'information vue comme un problème de codage sans perte

Dans cette partie, nous décrivons une méthodologie pour extraire l'information à partir de la mesure de similarité précédemment présentée. Nous décrivons un critère de longueur minimale de code pour un volume de données dans le but d'extraire une information d'importance et non redondante. Puis nous établissons quelques liens entre ce critère et la théorie débit-distorsion. Enfin, nous introduisons un algorithme sous-optimal qui minimise le critère et produit un index.

5.2.1 Extraction d'information d'une base d'objets

Nous rappelons que le décomposition LDM est une compression avec pertes en s'appuyant sur la fonction de structure introduite au §3.5.4 où chaque objet est traité séparément. Néanmoins dans le cadre de la théorie de Shannon, ces objets possèdent en moyenne de l'information commune et partagée. Par exemple, deux objets x et y peuvent mener à l'inférence de deux modèles représentant des propriétés identiques. Pour extraire l'information d'une base d'objets, il est souhaitable de supprimer les redondances. Ainsi nous pouvons coder un seul modèle représentant les propriétés de x et y au lieu de deux modèles redondants. Par conséquent, nous proposons de regrouper les modèles pour éliminer les redondances portées par les modèles. Cette approche est ni plus ni moins qu'un clustering de la base d'objets pour la création d'un index.
Soit l'ensemble $\{x_i\}_{i=1}^{q}$ une base de q objets. Pour avoir une extraction d'information optimale, nous souhaitons obtenir la représentation la plus compacte de la base d'objets. Un stratégie de codage consiste à partitionner la base en k sous-ensembles $\{S_i\}_{i\leq k}$ dans un premier temps. Ensuite, un modèle reflétant les propriétés d'un sous-ensemble S_i y est rattaché. Nous notons ce modèle $\mathcal{M}_{S_i}$. Nous pouvons diviser en trois parties la longueur de code de la base d'objets. La partition ou l'index peut être codé avec $qH(S)$ bits tel que :

$$p(S_i) = \frac{\mid S_i \mid}{q} \tag{5.24}$$

$$qH(S) = -\sum_{i=1}^{k} \mid S_i \mid \log p(S_i) \tag{5.25}$$

où $\mid S_i \mid$ est la taille de S_i. Cela revient à attribuer à chaque objet un numéro de groupe codé sur $-\log p(S_i)$ bits. Dans ce premier cas, nous considérons que la base admet une distribution uniforme. Dans un second temps, nous pouvons coder les modèles $\mathcal{M}_{S_i}$ avec $K(\mathcal{M}_{S_i})$ bits d'information. Enfin, comme nous connaissons l'appartenance des objets aux sous-ensembles, nous pouvons coder ceux-ci connaissant le modèle de leur groupe. Ainsi la longueur de code globale L s'exprime par :

$$L = \sum_{i=1}^{k} \left\{ - \mid S_i \mid \log p(S_i) + K(\mathcal{M}_{S_i}) + \sum_{x_j \in S_i} \{K(x_j \mid \mathcal{M}_{S_i})\} \right\} \tag{5.26}$$

Lors de l'extraction d'information, le but est alors de construire les modèles et la partition tels que le critère de la longueur de codage minimale suivant soit minimisé :

$$\{\hat{S}_i, \hat{\mathcal{M}}_{S_i}\}_{i\leq \hat{k}} = \arg \min_{k, S_i, \mathcal{M}_{S_i}} \sum_{i=1}^{k} \left\{ - \mid S_i \mid \log p(S_i) + K(\mathcal{M}_{S_i}) + \sum_{x_j \in S_i} \{K(x_j \mid \mathcal{M}_{S_i})\} \right\} \tag{5.27}$$

La décomposition, qui donne la longueur la plus courte et contient le moins de redondance, constitue l'extraction optimale. L'information extraite est alors contenue dans les modèles qui sont aussi les représentants de chaque groupe. L'index est quant à lui formé par le codage de l'appartenance de chaque objet à un cluster. Enfin, le troisième terme quantifie la dispersion de chaque sous-ensemble par rapport au modèle central. Le critère que nous souhaitons minimiser concerne la longueur de code globale de l'ensemble d'objets. Cependant, il s'opère un compromis entre l'information extraite et l'information sans

intérêt. L'information extraite est portée par l'index et les modèles, tandis que l'autre information représente la distorsion.

5.2.2 Liens avec la théorie débit-distorsion

Nous tenons à faire remarquer que le critère précédent s'apparente à une compression avec pertes en prenant un compromis particulier. Nous avons déjà fait remarquer ce cas de figure pour le principe LDM (cf. §3.5.4) réinterprété par la fonction de structure. En effet, nous souhaitons une double représentation qui est d'un point de vue sans perte et d'un autre point de vue avec pertes.

Soit X une variable aléatoire prenant ses valeurs dans l'ensemble d'objets $\{x_i\}_{i=1}^{q}$. Prenons $p(X)$ la distribution de ces objets. Si nous voulons transmettre tour à tour ces objets de manière indépendante, il faut dans un premier temps transmettre l'ensemble des modèles $\mathcal{M}_{S_i}$. Ensuite, pour chaque objet nous transmettons l'indice du modèle et la description de l'objet sachant le modèle. Ainsi la longueur moyenne L_m de transmission par objet, dans le cas où n réalisations sont transmises, s'exprime par :

$$L_m = \sum_{i=1}^{k} \left\{ \frac{K(\mathcal{M}_{S_i})}{np(S_i)} + \sum_{x_j \in S_i} p(x_j)\{K(x_j \mid \mathcal{M}_{S_i}) - \log p(S_i)\} \right\} \tag{5.28}$$

Si k est fixé et que n tend vers l'infini, nous remarquons que la longueur moyenne nécessaire à transmettre les modèles tend vers 0. Cependant, les modèles gardent leur importance dans l'expression des complexités conditionnelles. Par conséquent, la longueur de code moyenne s'approxime par :

$$L_m \approx \sum_{i=1}^{k} \sum_{x_j \in S_i} p(x_j)\{K(x_j \mid \mathcal{M}_{S_i}) - \log p(S_i)\} \tag{5.29}$$

Nous introduisons la fonction d'assignement déterministe représentée par la probabilité conditionnelle $p(S \mid X)$. Ainsi la longueur L_m peut se réécrire :

$$\begin{aligned} L_m &\approx \sum_{s,x} p(x)p(s \mid x)\{K(x \mid \mathcal{M}_s) - \log p(s)\} && (5.30)\\ &\approx \sum_{s,x} p(s,x) \log \frac{p(s \mid x)}{p(s)} + \sum_{s,x} p(s,x)K(x \mid \mathcal{M}_s) && (5.31)\\ &\approx I(X,S) + \mathbb{E}_{X,S}\left[K(X \mid \mathcal{M}_S)\right] && (5.32) \end{aligned}$$

Nous rappelons que dans le cadre de la théorie de Kolmogorov, la distorsion engendrée par le codage d'un objet x est donnée par la quantité $-\log p(x \mid \mathcal{M})$. Si nous identifions le terme $K(x \mid \mathcal{M})$ à $-\log p(x \mid \mathcal{M})$, alors le second terme de l'Eq. 5.32 est une mesure de distorsion moyenne. Dans ce contexte, le critère de longueur minimale (cf. Eq. 5.27) s'exprime sous la forme :

$$\min_{S_i, \mathcal{M}_{S_i}} L_m \approx \min_{p(s|x), \mathcal{M}_s} I(X,S) + \mathbb{E}_{X,S}\left[K(X \mid \mathcal{M}_S)\right] \tag{5.33}$$

Pour k fixé, il y a équivalence entre les deux minimisations précédentes puisque déterminer les ensembles S_i correspond exactement à calculer une fonction déterministe d'assignement $p(S \mid X)$. Donc la seconde minimisation englobe la première et est plus générale. La difficulté réside dans la détermination des modèles $\mathcal{M}_s$ qui correspondent à des centroïdes dans le cadre de la théorie débit-distorsion. En définitif, notre critère LDM (cf Eq. 5.28) est un cas particulier d'une minimisation de la fonction débit-distorsion, pour un paramètre de compromis égal à 1.
En opérant dans le sens inverse, il est logique de définir le critère débit-distorsion avec l'aide d'un multiplicateur de Lagrange β :

$$\min_{p(s|x),\mathcal{M}_s} I(X,S) + \beta \mathbb{E}_{X,S}\left[K(X \mid \mathcal{M}_S)\right] \tag{5.34}$$

Alors la longueur de code L_m dépend de β et prend une nouvelle forme :

$$L_m = \sum_{i=1}^{k} \left\{ \frac{K(\mathcal{M}_{S_i})}{qp(S_i)} + \sum_{x_j \in S_i} p(x_j)\{\beta K(x_j \mid \mathcal{M}_{S_i}) - \log p(S_i)\} \right\} \tag{5.35}$$

Par conséquent, ce nouveau critère permettrait de contrôler la quantité d'information extraite en réglant le paramètre β.
Nous terminons par établir le lien qui unit ce critère au principe IB. Si nous faisons l'hypothèse radicale que $K(X \mid \mathcal{M}) = -\log p(X \mid \mathcal{M})$, alors le second terme de distorsion de l'Eq. 5.34 correspond à $H(X \mid \mathcal{M})$. La minimisation de ce terme implique la maximisation du terme $H(X) - H(X \mid \mathcal{M}) = I(X, \mathcal{M})$. Par conséquent, nous maximisons l'information mutuelle entre le signal et les modèles. Cette condition ressemble fortement au principe IB, où le terme $I(\tilde{X}, \mathcal{M})$ est maximisé. Cependant les deux approches différent. Dans le principe IB, le but est d'extraire du signal X une partie de l'information qui va bien avec les modèles. Cette vision implique la connaissance a priori des modèles. Dans cette dernière approche, nous essayons de construire directement les modèles tels que le signal partage beaucoup d'information avec ceux-ci tout en limitant l'information extraite. L'information d'importance reste portée par les modèles.
Les liens présentés confortent le fait que le critère de longueur minimale proposé permet une extraction d'information optimale. D'autre part, nous avons établi les liens pour un k fixé. Cependant, faire varier le paramètre k dans le cas d'assignements déterministes, permet d'estimer une courbe débit-distorsion comme pour l'algorithme k-moyennes.

5.2.3 Procédure d'optimisation

La minimisation du critère Eq. 5.27 est très complexe. Nous proposons une méthode sous-optimale dans le but de minimiser ce critère. Nous introduisons tout d'abord le meilleur modèle $\mathcal{M}_S$ d'un ensemble S comme le modèle caractérisant au mieux les propriétés de l'ensemble. Cela se traduit en longueur de code par :

$$\mathcal{M}_S = \arg\min_{\mathcal{M}} \{K(\mathcal{M}) + \sum_{x \in S} K(x \mid \mathcal{M})\} \tag{5.36}$$

Déjà ce simple problème est difficilement résolvable. Nous donnons une procédure simplifiée et sous-optimale pour obtenir $\mathcal{M}_S$. Tout d'abord, nous inférons pour chaque objet

du sous-ensemble, le meilleur modèle au sens du critère LDM. Ensuite parmi cet ensemble de modèles, nous sélectionnons le meilleur et le plus représentatif, tel que :

$$\mathcal{M}_S = \min_{\mathcal{M}\in\{\mathcal{M}_x, x\in S\}} \{K(\mathcal{M}) + \sum_{x\in S} K(x \mid \mathcal{M})\} \tag{5.37}$$

Par conséquent, connaissant une partition $\mathcal{P}_k = \{S_1, ...S_k\}$ de l'ensemble d'objets, il est possible d'inférer le meilleur modèle pour chaque sous-ensemble par l'Eq. 5.37. Ces minimisations locales permettent de minimiser le critère globale.
Cependant, il nous faut calculer la partition de l'ensemble qui nous donnera l'index. Prenons le nombre de sous-ensembles k fixé. Chaque élément x admet une décomposition optimale en deux parties, tel que nous puissions calculer la matrice des distances δ entre chaque paire. Ensuite, nous appliquons l'algorithme de groupage hiérarchique sur cette matrice de distance pour regrouper deux à deux les éléments. Nous considérons la distance complète dc entre deux ensembles S et V pour construire l'arbre hiérarchique :

$$dc(S, V) = \max_{x\in S, y\in V} \delta(x, y) \tag{5.38}$$

Nous rappelons que la distance δ est une fonction de longueur de code conditionnelle. Ainsi, en prenant la distance maximale, deux ensembles sont proches si les éléments de l'un peuvent être codé efficacement connaissant l'autre. D'autre part, étant donné que nous utilisons la similarité en deux parties, nous cherchons à regrouper les ensembles dont les modèles sous-jacents sont proches et véhiculent des propriétés similaires. Par conséquent, en regroupant les objets similaires, nous forçons les sous-ensembles S à obtenir de meilleurs modèles.
Enfin, le paramètre k doit être sélectionné. L'arbre hiérarchique obtenu par l'algorithme d'agglomération renferme q partitions différentes dont le nombre de sous-ensembles varie de 1 à q. Nous notons $\mathcal{P}_k$ la partition qui contient k sous-ensembles. Le critère de compression maximale est restreint dans ce cas précis à :

$$\left\{\hat{k}, \{\mathcal{M}_{S_i}\}_{i\leq\hat{k}}\right\} = \arg \min_{k, S_i\in\mathcal{P}_k, \mathcal{M}_{S_i}} \sum_{i=1}^{k} \left\{ - \mid S_i \mid \log p(S_i) + K(\mathcal{M}_{S_i}) + \sum_{x_j\in S_i} \{K(x_j \mid \mathcal{M}_{S_i})\} \right\} \tag{5.39}$$

En comparant chaque partition, nous sommes capables de déterminer celle qui induit le moins de redondance pour l'extraction d'information. De plus la partition est un index de la base d'objet fondé sur l'information pertinente extraite.

5.2.4 Algorithme

Nous présentons sous forme de pseudo-code dans l'Algorithme 4 la procédure d'optimisation présentée au §5.2.3. Cette représentation permet de clarifier les entrées, les sorties de l'algorithme et les étapes majeures de l'optimisation. Les deux étapes majeures sont les codages indépendant et conjoint qui correspondent respectivement à l'extraction d'information et au clustering.

5.2.5 Avantages de l'indexation fondée sur le codage

Dans le cas où il existe un codeur universel en deux parties (C_1, C_2) qui permet d'extraire les modèles d'une base d'objets, la méthodologie précédente peut être appliquée.

Entrées : les données $\{x_i\}_{i=1}^{q}$ et le codeur en deux parties (C_1, C_2)
Sorties : le nombre de clusters $\hat{k}$, le clustering $\mathcal{P}_{\hat{k}}$ et les modèles $\{\mathcal{M}_S\}_{S\in\mathcal{P}_{\hat{k}}}$
Extraction d'information et codage indépendant
for $i = 1$ to q **do**
  $\{x_i \mid \mathcal{M}_{x_i}, \mathcal{M}_{x_i}\}$ = codage avec (C_1, C_2)
end for
Clustering et codage conjoint
calcul de la matrice des distances $\{DNSC(x_i, x_j)\}_{0\leq i,j\leq q}$
$\{\mathcal{P}_i\}_{i=1}^{q}$ = groupage hiérarchique($\{DNSC(x_i, x_j)\}_{0\leq i,j\leq q}$)
for $i = 1$ to q **do**
  for $S \in \mathcal{P}_i$ **do**
    $\mathcal{M}_S = \arg\min$ longueur de code de S (cf. Eq. 5.37)
  end for
  $L(\mathcal{P}_i)$ = longueur de code totale dépendant de $\{\mathcal{M}_S\}_{S\in\mathcal{P}_i}$
end for
Seléction du code
$\{\hat{k}, \mathcal{P}_{\hat{k}}, \{\mathcal{M}_S\}_{S\in\mathcal{P}_{\hat{k}}}\} = \arg\min L(\mathcal{P}_i)$

Algorithm 4 – Algorithme représentant la procédure d'optimisation pour le codage et l'indexation conjointes d'une base d'objets.

Il suffit d'approximer les longueurs de complexité de Kolmogorov par les longueurs de code obtenues.
Pour accélérer les calculs il est possible d'utiliser la distance DNSC au lieu de la distance δ, puisqu'il n'est pas nécessaire de refaire une inférence de modèle pour chaque calcul de la distance. Enfin, il est possible d'utiliser d'autres algorithmes pour calculer la partition de l'ensemble à partir de la matrice de distance.
En résumé, nous avons présenté une méthode générale (cf. Figure 5.1) permettant de compresser une base d'objets sans aucune perte. Cette méthode compresse la base entière en éliminant les redondances entre objets. De plus, elle permet d'intégrer un index dans le code. Cet index et les modèles constituent l'information d'importance extraite. Cette méthode n'est certes pas optimale, mais elle vise à s'approcher au mieux du minimum tout en construisant les caractéristiques de la base.
Cette méthodologie est inspirée des techniques classiques d'extraction de primitives, de réduction de dimensionnalité et de clustering pour la construction d'un index. L'avantage de notre méthodologie est qu'elle extrait l'information d'importance de manière non-supervisée contrairement à l'extraction des primitives. Du fait de la décomposition en deux parties à longueur minimale, nous savons que les propriétés pertinentes du signal sont portées par le modèle. Notre méthodologie permet d'obtenir une représentation unique d'un ensemble d'objets qui soit compacte et parcimonieuse, et qui permette une fouille par le contenu informationnel (cf. §5.1.4).
L'inconvénient majeur d'une telle méthode est la construction de la matrice de distance. Si n est le nombre d'objets de l'ensemble, notre méthodologie implique une complexité en $O(n^2)$. Nous proposerons par la suite (cf. §6.2.4.2) une solution pour contourner ce problème et réduire la complexité de calculs.

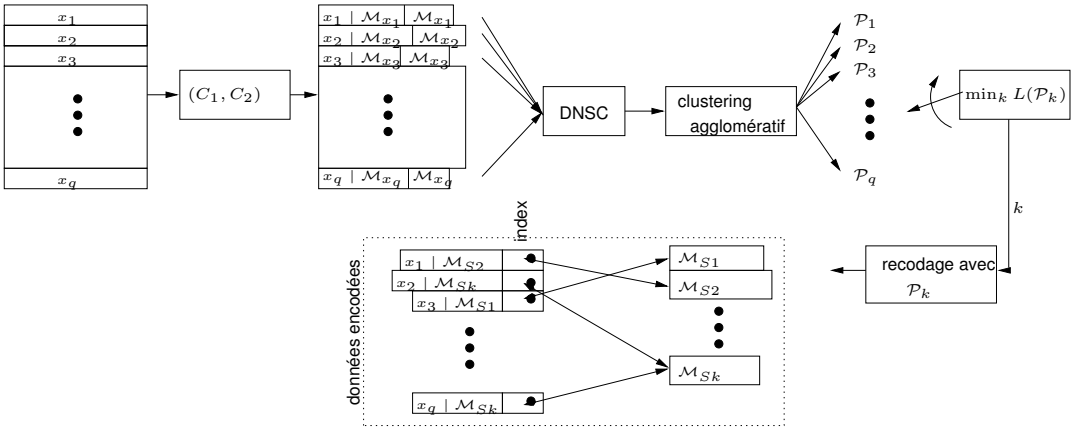


FIG. 5.1 – Ce diagramme décrit la méthodologie pour conjointement coder et créer un index d'une base de données. Premièrement, chaque objet est codé indépendamment à l'aide d'un codeur en deux parties. Puis, la matrice de distance DNSC est calculée entre chaque pair. Un algorithme d'agglomération hiérarchique est appliqué à la distance pour obtenir plusieurs partitions ou index qui sont presque optimaux. Enfin, fondée sur le critère de longueur minimale, la partition qui compresse au mieux la base entière est sélectionnée. Par un recodage de la base, l'index est inclus dans le code de l'ensemble.

5.3 Description d'un codeur sans perte pour STIS

Nous voulons appliquer la méthode générale précédente pour la compression et l'indexation de STIS. Nous nous concentrerons encore à caractériser les événements spatio-temporels contenus dans les STIS pour permettre la fouille de données. Pour appliquer la méthode, nous devons nous munir d'un codeur sans perte en deux parties pour la compression des STIS. Nous présentons dans cette partie un codeur de structures spatio-temporelles inspiré du codeur JPEG-LS (Weinberger et al., 2000).

5.3.1 Codeur sans perte pour les STIS

Un schéma général de codage en deux parties est présenté par Popat (1997). La Figure 5.2 donne une vue générale des différents modules qui composent le codeur. Comme nous souhaitons un codeur en deux parties, celui-ci fonctionne en deux passes. La première passe consiste à estimer le meilleur prédicteur et le meilleur arbre de contextes statistiques qui représentent les propriétés du signal. La seconde passe, consiste à coder l'erreur de prédiction en exploitant au mieux les contextes statistiques avec l'aide d'un codeur arithmétique adaptatif. Nous obtenons un codeur de type MAP présenté au §3.3.3, où les caractéristiques sont transmises séparément. Le schéma de décodage est symétrique et il est consistant avec le codage quand la prédiction et le décodage contextuel utilisent des voisinages causaux. Dans un premier temps, nous présentons comment inférer des prédicteurs causaux à partir des champs aléatoires autobinomiaux. Ensuite, nous présentons la modélisation des contextes statistiques fondée sur les gradients spatio-temporels quantifiés. Finalement, nous combinons ces deux modules pour satisfaire le principe LDM et obtenir un codeur universel en deux parties.
Dans le schéma de codage, nous n'utilisons pas de compensation de mouvements puisqu'il n'y en a pas à proprement parlé. Par contre, nous observons des variations temporelles de luminance qui sont mieux estimées par des prédictions spatio-temporelles. Nous

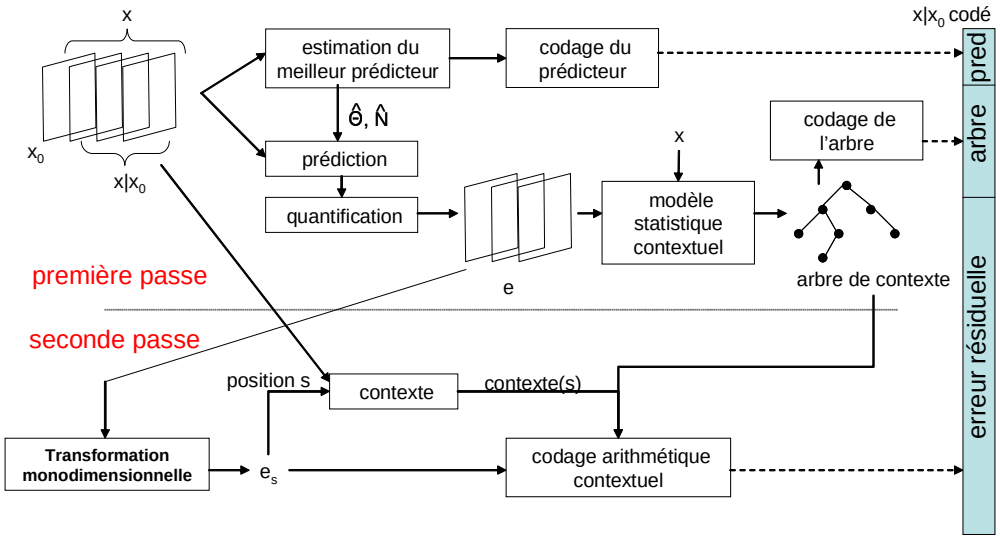


FIG. 5.2 – Schéma de codage sans perte en deux parties. Une partie correspond aux prédicteurs et à l'arbre des contextes statistiques. La seconde partie correspond à l'erreur résiduelle de prédiction.

supposons que les STIS sont monospectrales pour simplifier la méthode de codage.

5.3.2 Prédiction et champs autobinomiaux

Dans ce chapitre, nous nous appuyons sur les champs aléatoires autobinomiaux pour calculer les prédicteurs. Nous rappelons, que les champs autobinomiaux (cf. §3.1.5.2) décrivent les dépendances spatio-temporelles des STIS. De plus, un pixel se calcule comme une combinaison de ses voisins causaux et anti-causaux. Nous avons signalé que dans le but de faire une prédiction, les pixels doivent être estimés à partir d'un voisinage causal pour garantir une reconstruction parfaite. Nous revenons quelques instants à la définition des processus markoviens donnée au §3.1.3. Soit X_s la variable aléatoire associée au site s du champ aléatoire. Si les dépendances sont causales, X_s dépend seulement des pixels codés ou décodés dans le sens de lecture des sites. Nous restreignons les dépendances markoviennes des champs autobinomiaux à se faire avec les pixels causaux. Nous décidons arbitrairement de lire les champs ligne par ligne, puis image par image. Ainsi, nous explorons d'abord spatialement le champ avant de l'explorer temporellement. Par conséquent, nous modifions la définition de η (cf. Eq. 3.28) qui modélise les dépendances :

$$\eta = \theta_0 + \sum_{r \in N} \theta_r \frac{X_{s-r}}{\mathcal{G}} \tag{5.40}$$

Comme les demi-voisinages considérés sont anticausaux, nous exploitons les dépendances des voisins opposés. D'autre part nous approximons le modèle par un système linéaire comme au §4.3.1. Nous obtenons le système linéaire suivant en réarrangeant les positions données par les indices $a(s)$ et $b(r)$:

$$\vec{d} = G\vec{\theta} - \vec{e} \tag{5.41}$$

Le vecteur $\vec{d}$ et la matrice G sont définis tels que :

$$\vec{d}_{a(s)} = -\log\left(\frac{\mathcal{G}}{x_s} - 1\right) \tag{5.42}$$

$$G_{a(s),b(r)} = \begin{cases} 1 & \text{, if } r = 0 \\ \frac{X_{s-r}}{\mathcal{G}} & \text{, if } r \in N \end{cases} \tag{5.43}$$

Quand nous voulons prédire un pixel à partir de ses voisins précédemment traités et d'un vecteur de paramètres estimé $\hat{\theta}$, il suffit d'employer la procédure de prédiction suivante en deux étapes :

$$\hat{d}_{a(s)} = \sum_r G_{a(s),b(r)} \hat{\theta}_{b(r)} \tag{5.44}$$

$$\hat{x}_s = \lceil \frac{\mathcal{G}}{e^{-\hat{d}_{a(s)}} + 1} \rceil \tag{5.45}$$

où $\lceil . \rceil$ est un opérateur non linéaire donnant l'entier le plus proche. Chaque ligne $G_{a(s),.}$ de la matrice est composée des pixels précédemment traités à cause du voisinage employé. Par conséquent, les pixels peuvent être prédits séquentiellement les uns après les autres dans l'ordre donné par la fonction $a(s)$.
Dans ce schéma de prédiction, seuls les sites intérieurs du champ peuvent être prédits des pixels causaux voisins. En effet, les pixels qui sont sur les bords ne possèdent pas de voisins dans certaines directions. Pour surmonter ce problème, nous ne prédisons pas la première image du champ tridimensionnel, et choisissons de prédire les images successives. En l'occurrence, les pixels de la première image sont codés ou transmis avec d'autre moyens. Cependant, les pixels sur les bords des images suivantes ne peuvent pas être prédits efficacement non plus. Nous adoptons une extension artificielle du champ sur les bords à problème pour que chaque site appartenant au champ initial possède tous ses voisins causaux. L'étape d'extension s'opère à chaque instant dans le but de prédire efficacement l'image suivante sans perdre en efficacité dans la prédiction des bords. Nous présentons la procédure d'extension dans la Figure 5.3, où nous ajoutons des pixels sur les bords en leur attribuant la valeur du site le plus proche. Cette procédure d'extension respecte la causalité dans le but d'avoir une prédiction consistante lors du décodage. Nous préconisons de coder les pixels de la première image avec un codeur sans perte du type JPEG-LS. Dans le cadre de codage vidéo, cette étape correspond au codage des images intra.
La prédiction est valable tant que les paramètres sont connus. Nous proposons alors une méthode pour déterminer les paramètres de prédiction et les coder.

5.3.3 Apprentissage des prédicteurs dans le cadre LDM

Dans le but d'obtenir les paramètres de prédiction, nous utilisons une méthode d'apprentissage pour déterminer et coder un ensemble de prédicteurs.
Nous considérons un champ aléatoire X défini sur une grille de taille $l_x \times l_y \times l_t$. Nous prenons une base d'apprentissage qui correspond à une collection de plusieurs réalisations de ce champ $\{x\}$. Nous utilisons cette base pour calculer les prédicteurs les plus représentatifs de cette collection.
Comme nous l'avons signalé précédemment, la première image du champ ne peut être

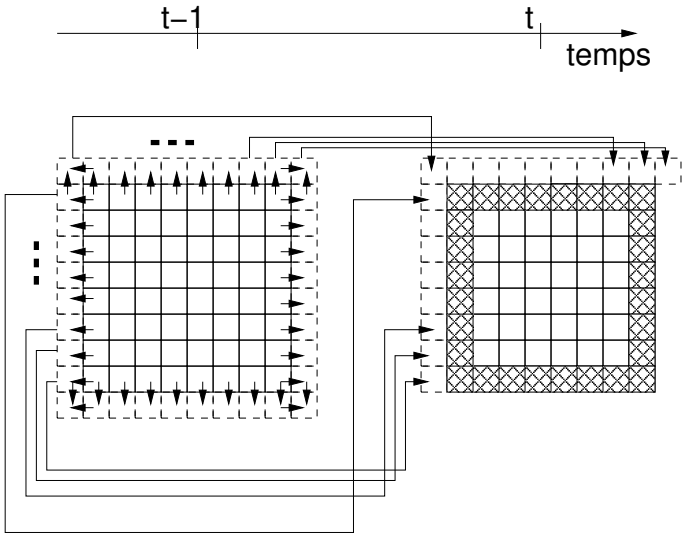


FIG. 5.3 – La méthode d'extension consiste à créer de nouveau pixels sur les bords à problème. Les pixels sont simplement copiés des pixels du champ initial les plus proches et déjà traité dans le sens de lecture ou de codage. Les bords à problème sont représentés par les pixels hachurés. Les pointillés représentent l'extension. L'opération de copie est symbolisée par un flèche qui va d'un site du champ vers un site de l'extension.

prédite. Nous appelons W ce champ aléatoire de taille $l_x \times l_y$. Ainsi, le champ aléatoire qui va nous permettre d'inférer les prédicteurs est donné par $X \mid W$ qui est défini sur une grille Ω de taille $l_x \times l_y \times (l_t - 1)$. Si le voisinage N est fixé et la taille de θ est k, le principe LDM s'écrit pour chaque réalisation :

$$\hat{\theta}(x \mid w) = \arg\min_{\theta} \{-\log p(x \mid w, \theta) + \frac{k}{2} \log (\frac{\mid \Omega \mid}{2\pi})\} \tag{5.46}$$

Comme le voisinage est fixé et que la taille du champ est fixé, le deuxième terme ne varie pas avec Θ. Par conséquent, c'est une estimation par Maximum de Vraisemblance. Cependant, nous rappelons que le second terme représente la longueur de code nécessaire à représenter les paramètres estimés. Alors il est raisonnable de penser que l'ensemble $\{\hat{\theta}(x \mid w)\}$ peut être codé avec $\frac{k}{2} \log (\frac{|\Omega|}{2\pi})$ bits par réalisation. Par des méthodes de construction de dictionnaire, nous pouvons identifier les mots du dictionnaire par un nombre de bits prédéterminé et ainsi limiter le codage des paramètres aux limites données par le principe LDM. Supposons que chaque mot du dictionnaire ai la même probabilité d'être utilisé, alors chaque mot peut être identifié idéalement avec $\log m$ bits, où m est le nombre de mots dans le dictionnaire $\mathcal{D}$. Si nous faisons en sorte que ce nombre soit strictement égal au nombre de bits théoriques nécessaire à coder les paramètres, nous obtenons les égalités suivantes :

$$\exists \quad 0 \leq \epsilon < 1, \quad \log (m - \epsilon) = \frac{k}{2} \log (\frac{\mid \Omega \mid}{2\pi}) \tag{5.47}$$

$$m = \lceil 2^{\frac{k}{2} \log (\frac{|\Omega|}{2\pi})} \rceil \tag{5.48}$$

$\lceil \rceil$ est l'opérateur non linéaire donnant l'entier supérieur le plus proche. L'égalité précédente détermine le nombre de mots que le dictionnaire doit contenir. Pour construire

le dictionnaire, nous proposons d'utiliser l'algorithme des k-moyennes pour réduire l'ensemble $\{\hat{\theta}(x \mid w)\}$ à m éléments représentatifs de la base d'apprentissage. Nous notons $\mathcal{D} = \{\tilde{\theta}\}$ cet ensemble de paramètres, de mots ou de prédicteurs. Par construction des k-moyennes nous espérons obtenir des paramètres représentatifs de la base d'apprentissage.
Cette réduction de l'espace $\{\hat{\theta}(x \mid w)\}$ est uniquement possible dans le cas où m est inférieur au nombre de réalisations de la base d'apprentissage. Si cette condition n'est pas respectée, il est possible d'adopter un condition plus souple qui correspond à avoir $m^{1/k}$ inférieur au nombre de réalisations. Cette condition revient à traité indépendamment chaque composante de θ et leur attribuer $\log(m)/k$ bits à chacune. Ensuite chaque composante peut être quantifiée par l'algorithme des k-moyennes en prenant $2^{\log(m)/k}$ centroïdes. Cette stratégie est valide à partir du moment où les composantes de θ sont indépendantes. C'est le cas idéalement, sinon le modèle employé est trop complexe et les paramètres ne sont pas des statistiques suffisantes.

5.3.4 Sélection et codage des prédicteurs

Nous nous intéressons à estimer le meilleur prédicteur et le coder, dans le but de prédire la STIS. Ainsi, le prédicteur est transmis d'un coté avec une longueur de code minimale et l'erreur de prédiction est transmise de l'autre côté.
Nous considérons une nouvelle réalisation y du champ aléatoire X qui n'est pas dans la base d'apprentissage. Nous estimons le meilleur prédicteur par une estimation MV restreinte au dictionnaire $\mathcal{D}$ qui s'exprime par :

$$\hat{\theta}(y \mid w) = \arg\min_{\tilde{\theta} \in \mathcal{D}} \{-\log p(y \mid w, \tilde{\theta})\} \tag{5.49}$$

Si y fait partie de l'ensemble d'apprentissage, la log-vraisemblance est augmentée par rapport à la meilleure estimée. Cependant, cette augmentation de la log-vraisemblance est petite en comparaison de la variance de l'erreur.
Le codage par dictionnaire présuppose que celui-ci soit connu du côté du codeur et du côté du décodeur. Ainsi, lors du codage de la réalisation $y \mid w$, le meilleur prédicteur est estimé et les paramètres correspondant sont codés par leur indice ou leur position dans l'ensemble numéroté $\mathcal{D}$. Cette étape constitue déjà une indexation des réalisations, en groupant les objets par les indices des mots du dictionnaire.
Dans cette modélisation du champ aléatoire, l'erreur de prédiction est un processus stochastique i.i.d. Du fait des approximations des paramètres, cette proposition n'est plus valable. Nous donnons par la suite une modèlisation statistique plus fine de l'erreur.

5.3.5 Modèle statistique de l'erreur résiduelle

Nous décrivons un modèle statistique de l'erreur fondé sur les contextes. Cette modélisation a été appliquée avec succès pour le codage sans perte des images (Weinberger et al., 2000; Wu & Memon, 1997). Dans un premier temps nous définissons l'erreur de prédiction, puis nous donnons la modélisation des contextes sous forme d'arbre de gradients.
Lors de la prédiction, un pixel central est estimé à partir de ces voisins et prend une valeur entière. L'erreur de prédiction e_s est définie comme la différence entre la prédiction et l'original, $e_s = x_s - \hat{x_s}$. Si les niveaux de gris possibles appartiennent à l'intervalle $[0, \mathcal{G}]$, il est évident que l'erreur appartient à un intervalle d'entiers restreint. Dans le

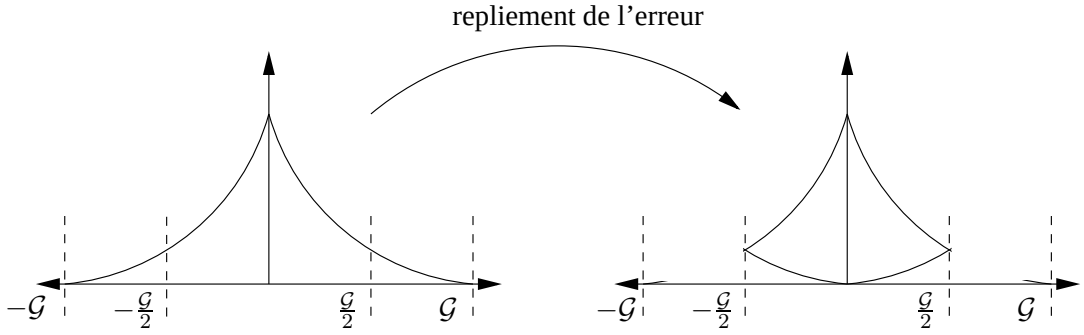


FIG. 5.4 – Ce graphe présente comment la probabilité de l'erreur est repliée vers le centre. Etant donné que la probabilité de l'erreur sur les bords est très faible, elle n'influe pas beaucoup sur la probabilité centrale que est très importante.

cas où le pixel prédit est $\hat{x_s}$, l'erreur de prédiction e_s appartient à l'intervalle d'entiers $[-\hat{x}_s, \mathcal{G} - \hat{x}_s]$. En conséquence, si nous définissons une variable aléatoire E_s prenant comme réalisations les valeurs e_s, l'espace des réalisations dépend de s. Cependant, nous voulons modéliser l'erreur par un processus dont les variables aléatoires prennent leurs valeurs dans un même espace. Ainsi, l'espace des réalisations ne peut être dépendant du site s et de la valeur de $\hat{x}_s$. Weinberger et al. (2000) préconisent de replier les bords de l'intervalle sur le centre. Puisque les valeurs de l'erreur sont centrées sur 0, les erreurs au bord de l'intervalle ont une très faible probabilité d'apparition et n'influent pas sur la distribution au centre. Ce repliement est possible, puisque connaissant $\hat{x}_s$, e_s ne peut prendre que $\mathcal{G}$ valeurs et l'intervalle $[-\mathcal{G}, \mathcal{G}]$ est trop grand pour décrire l'erreur. Ainsi, en contraignant e_s à être compris entre $-\mathcal{G}/2$ et $\mathcal{G}/2 - 1$ modulo $\mathcal{G}$, il est possible de retrouver sa vrai valeur connaissant $\hat{x}$. Cette opération de clippage consiste à replier les queues des distributions vers le centre. La Figure 5.4 exemplifie ce traitement. Nous donnons la définition de la fonction de clippage :

$$clip(e_s) = \mod (e_s + \frac{\mathcal{G}}{2}, \mathcal{G}) - \frac{\mathcal{G}}{2} \tag{5.50}$$

où $\mod (.,.)$ donne le reste de la division euclidienne.
Nous modélisons la probabilité d'erreur en fonction du contexte du pixel courant. D'après les études réalisées, les distributions sont fortement liées aux gradients environnants. Alors, nous définissons un contexte causal à partir du voisinage causal du pixel traité. Ce contexte est fondé sur les différences entre pixels voisins. Pour éviter la dissolution de contexte, ces différences sont quantifiées comme Wu & Memon (1997) le recommandent. Cette quantification a pour effet de diminuer le nombre de contextes possibles. Ainsi, lors de l'estimation des probabilités pour chaque contexte, le nombre de réalisations est suffisamment grand.
Dans le cadre de la compression des STIS, nous considérons les gradients exposées dans la Figure 5.5, qui correspondent aux gradients horizontaux, verticaux, diagonaux et temporels, représenté par le vecteur de contexte $\{\mid a-c \mid, \mid c-b \mid, \mid b-h \mid, \mid e-k \mid\}$. Ce contexte identifie les contours qui pourraient changer le comportement des distributions de l'erreur. Chaque différence est quantifié sur les intervalles suivants :$\{0, 1\}$, $\{2...5\}$, $\{6...9\}$, $\{10...14\}$, $\{15...\}$. Cette quantification diminue le nombre de vecteurs de contextes possibles, qui se réduit dans notre cas à 5^4 états différents. La quantification opérée revient à uniformiser la distribution des contextes sur tous les types d'images. Cette supposition

vient des a priori que nous avons sur la formation d'une image. Nous étendons cette supposition aux SITS. Néanmoins, la quantification devrait être différente et adaptée aux gradients temporels. Par souci de simplicité, nous laissons ce point comme une extension possible du codeur.
Pour chaque contexte, nous pouvons estimer une probabilité de l'erreur de prédiction. Pour évaluer ces probabilités, nous nous basons sur l'arbre de contextes introduit par Weinberger et al. (1996a) et présenté dans la Figure 5.6. Cet arbre est une représentation des contextes sous forme de regroupements hiérarchiques. Le premier niveau de l'arbre contient un certain nombre de noeuds où chaque noeud est un groupe de contextes discriminé par la première valeur du vecteur. Les noeuds du dernier niveau correspondent quant à eux à la totalité des états. Clairement, le vecteur de contexte identifie de manière unique le chemin de la racine vers l'état du contexte lui-même. Nous associons à chaque noeud de l'arbre, une distribution de l'erreur estimée par un histogramme. Ces distributions sont des distributions conditionnelles qui vérifient l'égalité suivante :

$$p(E_s \mid u) = \sum_{v:P(v)=u} p(E_s \mid v) \tag{5.51}$$

où u est un noeud intérieur et v est un de ses fils. L'arbre complet résultant est appelé arbre complet des contextes des gradients quantifiés. Ainsi lors du codage, nous connaissons la longueur de code nécessaire à coder l'erreur sachant le contexte. Cependant, pour avoir un codage optimal, il est nécessaire que les gains obtenus sur le codage de l'erreur ne soient pas perdus lors du codage de l'arbre des contextes statistique. Il n'est par exemple pas utile de conserver l'arbre complet, pour obtenir des gains de codage satisfaisant.
Soit, T un sous arbre de l'arbre complet, où chaque distribution est paramétrée avec d valeurs. Weinberger et al. (1996b) montrent que le principe LDM appliqué à cet modélisation statistique sous forme d'arbre se traduit par :

$$\min_{d,T} \{-\log p(e^n \mid T) + \frac{k(T)d}{2} \log n\} \tag{5.52}$$

où e^n est un réalisation du processus stochastique des erreurs, $k(T)$ est le nombre de feuilles et n est la taille du processus. Pour un champ aléatoire, n correspond au nombre de sites que contient la grille. Ainsi, un arbre trop complexe est pénalisé, puisqu'il requiert trop de bits.
Weinberger et al. (1996a) propose une méthode d'estimation presque optimale pour déterminer le sous arbre T. Cette méthode s'effectue pour une taille des paramètres d fixée. Popat (1997) propose une méthode plus élaborée pour optimiser ces deux paramètres conjointement. Pour des soucis de rapidité, nous privilégions la méthode de Rissanen et supposons que la distribution conditionnelle de chaque noeud est connue. La méthode consiste à élaguer l'arbre complet des contextes de gradients. A chaque noeud u est associé un coefficient d'efficacité $E_f(u)$. Ce coefficient calcul la perte entropique de codage quand l'erreur est encodée avec la distribution du noeud parent. Ce coefficient se calcul par :

$$E_f(u) = \sum_{e:cont(e)=u} -\log p(e \mid u) - (-\log p(e \mid P(u))) \tag{5.53}$$

où $\{e : cont(e) = u\}$ est l'ensemble des réalisations de l'erreur dont le vecteur de contexte détermine un chemin qui passe par le noeud u. $P(u)$ est le parent du noeud u. Ce coefficient se calcul pour tous les noeuds intérieurs de l'arbre complet. Rissanen propose

d'élaguer l'arbre en supprimant tous les noeuds dont le coefficient d'efficacité est plus petit qu'un seuil E_t. Pour obtenir un sous arbre consistant, si un noeud u est éliminé, le sous arbre dont la racine est u est aussi supprimé. Cette procédure vise à éliminer les noeuds qui ne produisent pas un gain de codage significatif ainsi réduisant la complexité de l'arbre. L'optimisation directe de l'Eq. 5.52 est très complexe. Aussi, Rissanen propose de restreindre la minimisation aux arbres obtenus par élagage. Le critère LDM se réécrit sous la forme suivante :

$$\min_{d,E_t} \{-\log p(e^n \mid T(E_t)) + \frac{k(T(E_t))d}{2} \log n\} \tag{5.54}$$

où $T(E_t)$ est le sous arbre obtenu en élaguant l'arbre complet avec le seuil E_t. Cette procédure vise à pénaliser les arbres avec trop de feuilles. Quand le seuil augmente, $T(E_t)$ tend vers le sous-arbre composé uniquement de la racine. Au contraire, quand E_t diminue, le sous arbre $T(E_t)$ tend vers l'arbre complet.
La dimension d des distributions conditionnelles reste à être discutée. Netravali & Limb (1980) montrent que la distribution exponentielle est un bon modèle pour la distribution de l'erreur. Cette distribution est paramétrée par σ tel que :

$$p(e \mid \sigma) = \frac{\exp(- \mid e \mid /\sigma)}{Z_\sigma} \tag{5.55}$$

C'est une distribution centrée sur 0 pour correspondre aux prédicteurs. Le terme de normalisation Z_σ permet de normaliser la distribution sur l'intervalle d'entiers $[-\mathcal{G}/2, \mathcal{G}/2 - 1]$. Pour chaque noeud, les paramètres de chaque distribution sont estimés :

$$2\hat{\sigma}_u^2 = \sum_{e:cont(e)=u} e^2 \tag{5.56}$$

Le terme de normalisation correspondant se calcule par :

$$Z_{\hat{\sigma}_u} = \sum_{e=-\mathcal{G}/2}^{\mathcal{G}/2-1} \exp(- \mid e \mid /\hat{\sigma}_u) \tag{5.57}$$

Ces distributions normalisées sont utilisées pour la minimisation du critère LDM et pour le calcul des coefficients d'efficacité. Nous avons confronté d'autres modèles de distributions, et la distribution exponentielle donne les longueurs minimales de code. Nous avons testé par exemple les gaussiennes, les histogrammes et les laplaciennes à deux paramètres. En résumé cet arbre contextuel statistique combiné à un codeur arithmétique permet de coder efficacement l'erreur résiduelle en deux parties en suivant le principe LDM.

5.3.6 Extraction d'information et mesure de similarité

Nous nous référons au système de codage en deux passes présenté dans la Figure 5.2. Nous avons présenté les moyens pour combiner les différents modules et obtenir un codeur sans perte en deux parties efficace. Le codage du prédicteur et de l'arbre, représente le modèle du signal ou l'information d'importance extraite. D'autre part, le signal d'erreur constitue la seconde partie du code qui est considérée comme purement aléatoire connaissant l'arbre de contexte. En l'occurrence, nous faisons en sorte que chaque bit

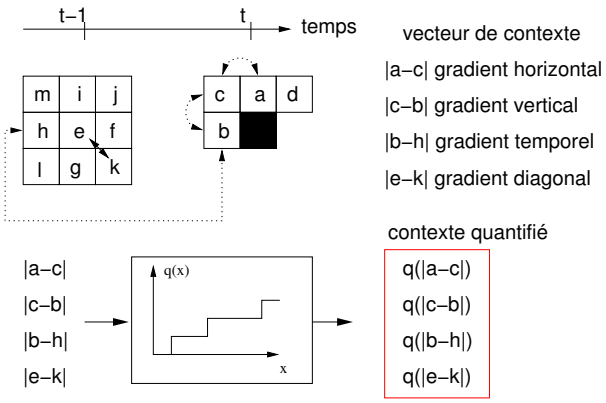


FIG. 5.5 – Le contexte d'un pixel, représenté en noir, est défini à partir des gradients avoisinants. Ces gradients sont quantifiés pour réduire le nombre total de contextes et éviter une dissolution de l'information conditionnelle. Dans le cadre de STIS, nous considérons les gradients horizontaux, verticaux, diagonaux et temporels pour définir le contexte.

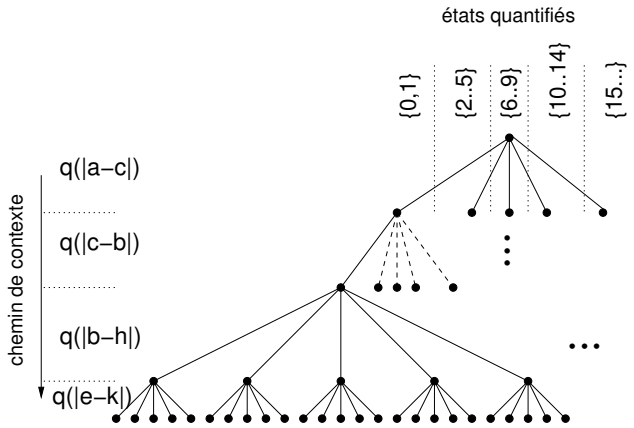


FIG. 5.6 – L'arbre complet correspondant aux contextes $\{q(|\ a-c\ |), q(|\ c-b\ |), q(|\ b-h\ |), q(|\ e-k\ |\ )\}$ est représenté. Chaque vecteur de contexte correspond à un chemin vers un des 4^5 contextes. A chaque noeud est associé une distribution de probabilité d'erreur.

numérique transmis contienne un bit d'information. C'est la définition d'un signal purement aléatoire. Par analogie avec les techniques classiques d'extraction d'information dans les images, nous pouvons considérer le prédicteur et l'arbre comme primitives du signal codé.
Ce schéma de codage en deux parties permet de définir la Distance Normalisée Suffisante de Compression (DNSC) entre deux signaux tridimensionnels x et y. Dans le calcul de cette distance, nous nous intéressons aux complexités conditionnelles $C_1(x \mid \mathcal{M}_y)$ et $C_1(y \mid \mathcal{M}_x)$. Prenons la mesure $C_1(x \mid \mathcal{M}_y)$. Cette quantité mesure la quantité de bits nécessaire à coder l'erreur de prédiction de x en prenant le prédicteur et l'arbre statistique contextuel obtenus avec y. L'union de ces deux objets, prédicteur et arbre, est par conséquent noté $\mathcal{M}_y$.
Finalement, notre codeur peut être utilisé pour le codage de la STIS vue comme un ensemble de structures spatio-temporelles en se basant sur la méthodologie présentée au §5.2.1. En effet, en codant indépendamment les structures spatio-temporelles, nous réduisons les redondances et extrayons l'information intrinsèque à chaque objet. Dans une seconde étape, en exploitant les similarités inter-objets, nous réduisons les redondances qui existent entre les événements spatio-temporels. Cette méthode permet d'augmenter la compression de toute la STIS tout en créant un index du contenu informationnel.
Nous terminons par signaler que le codeur présenté s'applique aux séries temporelles monospectrales. Néanmoins, il est possible de généraliser notre codeur à des séries temporelles multispectrales.

5.4 Résumé

Nous avons introduit la Distance Normalisée Suffisante de Compression (DNSC) pour comparer deux objets en utilisant l'information structurante des objets. Nous avons déduit cette distance de la DCN qui présente de très bons résultats pour le clustering de différents types d'objets, tels que le texte, les textures ou l'ADN. La DNSC est pertinente dans le cadre de l'extraction d'information, puisqu'elle intègre le principe LDM qui vise à extraire l'information d'intérêt.
Ensuite, nous avons présenté un critère général de codage d'un ensemble d'objets en exploitant les redondances inter-objets. Ce principe rentre dans le cadre LDM, où l'information est divisée en deux parties. D'autre part, nous montrons que la minimisation de la longueur de code s'apparente à la minimisation d'un couple débit-distorsion. Ce lien conforte le fait qu'une extraction d'information est opérée lors du codage conjoint de l'ensemble d'objet. Dans le cadre de la théorie débit-distorsion, nous montrons que le critère de longueur minimale s'apparente au principe IB puisque une troisième variable contenant l'information pertinente est introduite. Le critère proposé dans l'Eq. 5.27 ne peut être minimisé directement. Pour surmonter le problème, nous proposons une méthode d'optimisation fondée sur la DNSC et sur la décomposition de l'information en deux parties par un codeur universel. Cette méthode permet de construire un index des modèles. D'une part, cet index contient l'information pertinente extraite, d'autre part il est intégré dans le code en deux parties de l'ensemble d'objets.
Dans le but d'appliquer notre méthodologie d'extraction aux STIS, nous créons un codeur sans perte pour les signaux tridimensionnels. Nous exploitons l'expérience scientifique accumulée sur le codage d'image (Weinberger et al., 2000; Wu & Memon, 1997)

pour construire ce codeur universel. Plus le codage est efficace, plus le modèle sous-jacent explique bien les données codées.
Néanmoins, la couleur n'est pas prise en considération. En effet le codeur proposé est dédié aux séries monospectrales. Pour inclure l'information de couleur, comme cela est fait au §4.4.4, nous préconisons d'utiliser des transformées de décrorrélation spectrales, du type Karhunen-Loeve, comme modèle de l'information de couleur (Gueguen et al., 2005).

Troisième partie

Expérimentation et analyse des résultats

Chapitre 6

Validation et analyse des résultats

Dans ce chapitre, nous nous attachons à valider les méthodologies que nous avons introduites. Pour pouvoir les évaluer, nous les appliquons à des données que nous synthétisons. Ensuite, nous comparons les différentes méthodes à des méthodes d'extraction d'information de référence. Enfin, nous discutons des résultats d'extraction d'information obtenus sur la STIS ADAM.

6.1 Validation des méthodologies fondées sur le principe IB

Dans ce chapitre, nous mettons en évidence la pertinence des résultats obtenus sur des données synthétiques. Dans le but de valider les méthodologies fondées sur le principe IB et MIB, nous expérimentons ces algorithmes sur des données déjà classées. Ainsi, nous pouvons comparer les résultats de clustering aux classes existantes. Enfin, nous appliquons les deux méthodologies aux STIS et discutons des résultats.

6.1.1 Validation du clustering fondé sur le principe IB

Pour l'évaluation de la méthode fondée sur le principe IB (cf. §4.2), nous la comparons avec les algorithmes k-moyennes et AutoClass (Cheeseman et al., 1988). Pour obtenir une comparaison consistante, nous appliquons ces différentes méthodes de clustering sur des données synthétiques comme des signaux autorégressifs. Nous choisissons ce type de signaux pour faire le rapprochement avec les champs aléatoires de Gauss-Markov. Nous présentons deux objectifs dans cette validation. Dans un premier temps, nous montrons que notre méthodologie produit une classification non supervisée proche d'une classification initiale. D'un autre coté, nous montrons que notre méthodologie est adaptée à déterminer le nombre optimal de clusters, c'est à dire la quantité optimale d'information d'importance. Nous comparons notre méthodologie à AutoClass, puisque cette dernière technique de clustering permet aussi d'évaluer le nombre optimal de clusters.
Pour évaluer les différentes méthodes, nous générons des signaux aléatoires qui sont causalement autorégressifs. Ainsi, pour un jeu de paramètres θ de dimension k, un signal x est modélisé par :

$$x_n = e_n + \sum_{i=1}^{k} \theta_i x_{n-i} \tag{6.1}$$

où e_n est un bruit blanc gaussien de variance unitaire. Nous construisons plusieurs signaux à partir de ce modèle. Nous considérons les modèles des trois premiers ordres.

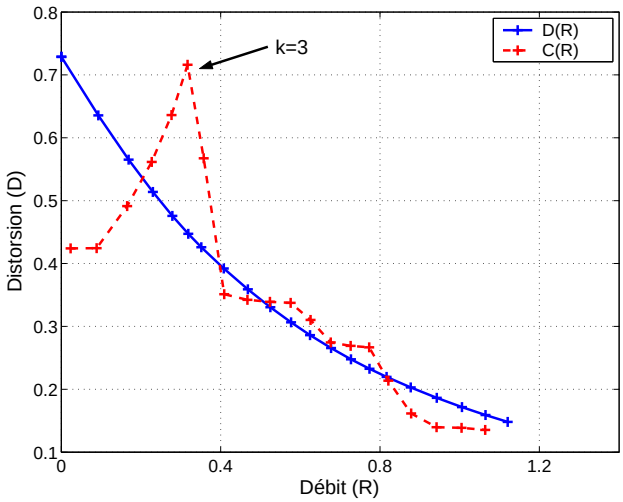


FIG. 6.1 – La courbe débit-distorsion et la courbure obtenues sur les signaux autorégressifs avec l'algorithme IB. Le maximum local détermine le nombre de clusters qui est trois dans ce cas.

Pour générer les différents paramètres, nous adoptons une distribution uniforme dans les trois espaces respectifs. Ainsi, nous constituons 3 classes de signaux dont la complexité des modèles varie entre 1 et 3 paramètres. Le fait que la distribution est uniforme dans chaque catégorie de modèle n'apporte pas d'information particulière ou caractérisant l'ensemble. Ainsi, il n'y a pas de raffinement du clustering au delà de la complexité des modèles.
Nous décrivons maintenant la procédure de comparaison. Dans un premier temps, nous appliquons la méthodologie fondée sur le principe IB, en prenant $\mathcal{M}$ comme variable d'intérêt et qui prend ses valeurs dans l'espace des modèles autorégressifs des trois premiers ordres. La Figure 6.1 présente la courbe débit-distorsion correspondante. Le premier maximum de la courbure correspond à 3 clusters qui équivaut au nombre de classes. Par ailleurs, nous observons d'autre maximaux correspondant à des sous classes. Dans un second temps, nous appliquons l'algorithme k-moyennes. Nous appliquons un sélection de modèles globale sur tous les signaux et trouvons que le modèle d'ordre deux convient le mieux à représenter les signaux. Ensuite, pour chaque signaux nous estimons les paramètres d'ordre deux avant de les grouper avec l'algorithme k-moyennes en considérant 3 groupes. Dans ce cas, nous forçons le nombre optimal de clusters. Enfin, nous appliquons l'algorithme AutoClass sur les mêmes primitives extraites en laissant l'algorithme déterminer le nombre de clusters.

Nous présentons dans le Tableau 6.1, les matrices de confusion entre les différents groupages et la classification initiale. Nous signalons que nous avons 100 échantillons par classe. Nous observons que notre méthode donne de meilleurs résultats en termes de confusion. En effet, les algorithmes k-moyennes et AutoClass n'arrive pas à extraire la structure des classes. Néanmoins, AutoClass réussit à déterminer que le nombre de classes optimal est trois. Ces résultats montrent l'intérêt du clustering fondé sur le principe IB pour extraire l'information pertinente. D'autre part, nous prouvons que cette méthodologie est adaptée à des modèles autorégressifs tels que les champs de Gauss-Markov.

	IB clustering			k-moyennes			AutoClass		
	clusters			clusters			clusters		
classes	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
1	**83**	16	1	**38**	41	21	83	10	7
2	27	**52**	21	30	**47**	23	89	7	4
3	7	36	**57**	34	41	**25**	63	31	6

TAB. 6.1 – Les matrice de confusion correspondant aux k-moyennes, AutoClass et au clustering IB. Les résultats montrent que les algorithmes k-moyennes et AutoClass n'exploitent pas l'information protée par les modèles en produisant des matrices confuses. Néanmoins, AutoClass détermine la présence de 3 classes.

6.1.2 Validation du principe MIB

Nous souhaitons démontrer l'avantage de la méthodologie fondée sur le principe MIB sur des données synthétiques. Pour cela nous considérons un signal aléatoire $Z(t)$ tel que :

$$Z(t) = (A + N_A)\cos\left((\Phi + N_\Phi)t\right) \tag{6.2}$$

où A, Φ sont deux variable qui prennent leurs valeurs dans des espaces discrets finis $\{a_1, a_2\}$ et $\{\phi_1, \phi_2\}$, respectivement. N_A et N_Φ sont deux bruits blancs gaussiens de moyennes nulles et de variances inconnues. Nous pouvons constater qu'il y a 4 grandes classes de signaux en combinant les différents états des variables A et Φ. Nous pouvons donc supposer que l'information d'importance est portée par ces deux variables. En effet, leur connaissance permet de donner les caractéristiques du signal sous-jacent. Alors, nous nous intéressons à la variable aléatoire $X = (A + N_A, \Phi + N_\Phi)$. Nous supposons que les distributions de A et Φ sont uniformes et que A et Φ sont indépendantes pour générer les réalisations synthétiques de $Z(t)$. Par une estimation fondée sur la transformation de Fourier, nous obtenons les réalisations de X, dont un exemple est présenté dans la Figure 6.2.

Nous voulons analyser le contenu informationnel de X en introduisant deux variables aléatoires Y_1, Y_2 prenant leurs valeurs dans $\{a_1, a_2\}$ et $\{\phi_1, \phi_2\}$. Ces deux variables d'intérêt sont liées à X par les distributions suivantes :

$$p(X \mid y_1) = p(A + N_A \mid y_1) \quad = \quad \mathcal{N}(y_1, 1) \tag{6.3}$$
$$p(X \mid y_2) = p(\Phi + N_\Phi \mid y_2) \quad = \quad \mathcal{N}(y_2, 1) \tag{6.4}$$

Nous utilisons la méthode du §4.3.2 pour calculer $p(Y_1 \mid X)$ et $p(Y_2 \mid X)$ à partir de $p(X \mid Y_1)$ et $p(X \mid Y_2)$. De plus nous admettons que X suit une distribution uniforme sur l'espace des réalisations considérées. Ainsi, nous pouvons évaluer les informations mutuelles $I(X, Y_1)$ et $I(X, Y_2)$. La première quantité nous renseigne sur l'information relative à l'amplitude contenu dans la variable Z. La seconde quantité nous renseigne sur l'information relative à la période contenue dans Z. Nous pouvons par conséquent quantifier ces deux types d'information.

Nous rappelons la définition des distorsions normalisées dans le critère MIB normalisé :

$$d^i = \frac{D^i}{I(X, Y_i)} \tag{6.5}$$

En prenant ces distances normalisées dans le principe MIB, aucun type d'information n'est privilégié a priori et seuls les paramètres de compromis pourrons pondérer l'im-

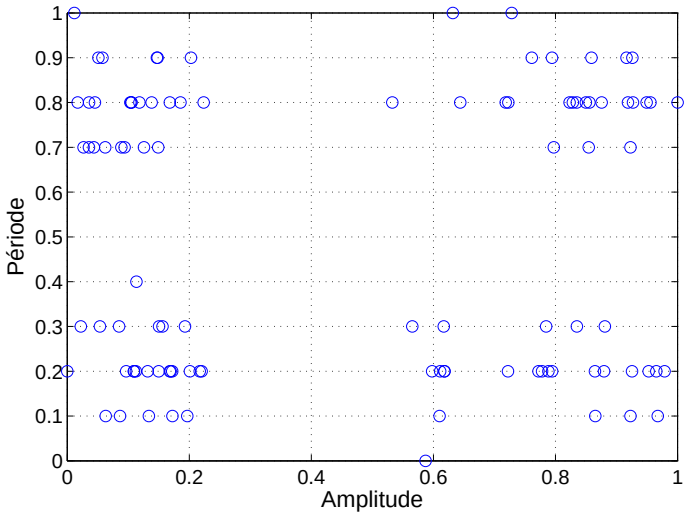


FIG. 6.2 – Les estimations correspondant aux amplitudes A et aux périodes Φ de $Z(t)$ sont présentés. Nous observons 4 groupes distincts qui correspondent aux 4 états qui ont permis de générer les signaux $z(t)$.

portance des informations. Nous appliquons l'algorithme et traçons la surface débit-distorsion (R, d^1, d^2) dans la Figure 6.3. Nous obtenons les courbes débit-distorsion des deux problèmes IB en prenant soit $\beta_1 = 0$, soit $\beta_2 = 0$. Cela revient à supposer que Y_1 (ou Y_2) véhicule seule l'information d'importance. Ces courbes déterminent les limites sur les bords de la surface et sont représentées dans la Figure 6.4.

Nous présentons quelques résultats de clustering déterministes pour différentes valeurs des paramètres β_1, β_2. Pour se donner une idée représentative des clusters, nous comparons les groupes obtenus aux 4 classes initiales correspondants aux états $\{(a_1, \phi_1), (a_2, \phi_1), (a_1, \phi_2), (a_2, \phi_2)\}$. Pour comparer les classes aux groupes des clusterings, nous donnons les matrices de confusion qui calculent le nombre d'éléments commun à un groupe et une classe. Quand la matrice de confusion est carrée et diagonale, le clustering est optimal. En effet, celui-ci correspond exactement à la classification que l'on souhaite retrouver. Plusieurs matrices de confusion sont données dans la Table 6.2. Etant donné que les distances employées sont normalisées, les paramètres de compromis sont absolus, c'est-à-dire qu'ils représentent le poids absolu d'un type de paramètre. Tout d'abord nous notons à la première et seconde ligne du tableau que les clusterings obtenus reflètent l'information d'intérêt. En effet, en prenant des paramètre de compromis égaux $\beta_1 = \beta_2$, nous donnons autant d'importance aux informations véhiculées par Y_1 et Y_2. En outre, l'algorithme MIB est capable de retrouver cette structure de classes correspondant à 4 états. Dans la troisième ligne de la Table 6.2, nous privilégions l'information relative à Y_2 en prenant $\beta_2 > \beta_1$. Ainsi, nous obtenons 2 clusters dont un mélange les classes $(a_1, \phi_1), (a_2, \phi_1)$ et l'autre mélange les états $(a_1, \phi_2), (a_2, \phi_2)$. C'est cohérent car l'information portée par l'amplitude est atténuée. A la troisième ligne du tableau, nous remarquons le cas inverse où l'information portée par l'amplitude est privilégiée. Enfin, les dernières lignes de la table, nous donne des cas intermédiaires.

En conclusion, nous mettons en évidence dans cet exemple, que le principe MIB permet d'extraire différents types d'information indépendants. Ainsi, nous pouvons quantifier et qualifier l'information contenue dans une variable aléatoire. En jouant sur les paramètres de compromis, nous sommes capables de privilégier tel où tel type d'information. En

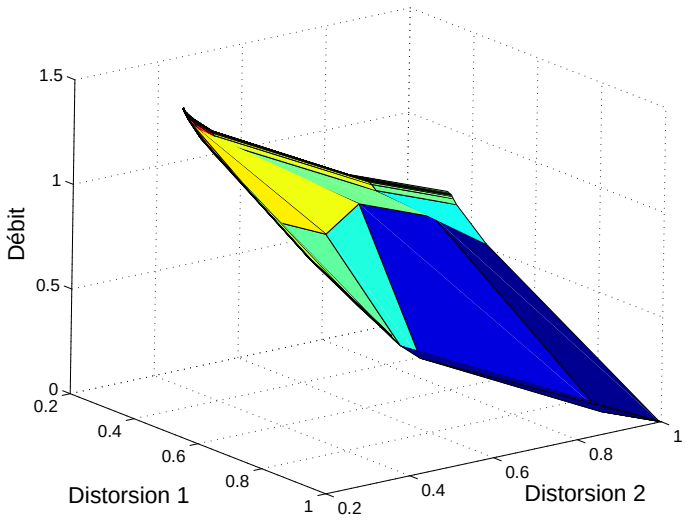


FIG. 6.3 – Cette surface débit-distorsion est obtenue en prenant la courbe paramétrique $(R(\beta_1, \beta_2), d^1(\beta_1, \beta_2), d^2(\beta_1, \beta_2))$. Nous observons que cette courbe est quasiment symétrique.

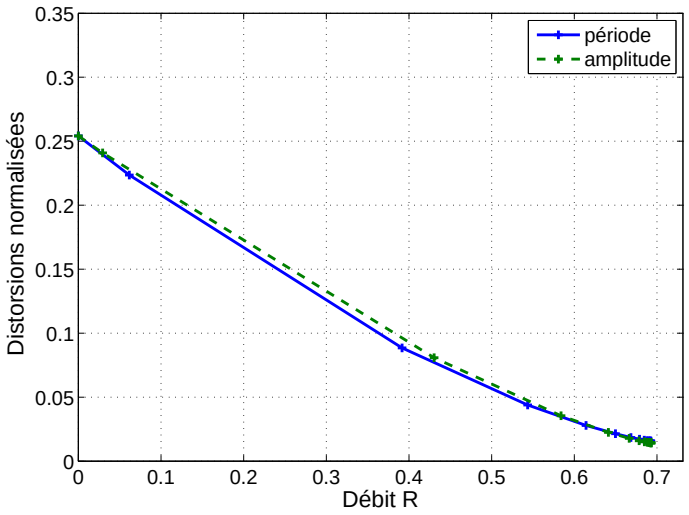


FIG. 6.4 – Ces deux courbes sont obtenues pour les cas limites du principe MIB. Elles correspondent à deux problèmes IB résolus indépendamment. Les distances normalisées sont représentées en parallèle pour comparer l'évolution des deux courbes. Nous remarquons dans notre cas, qu'elles ont un comportement similaire.

β_1	β_2	nombre effectif de centroïdes	matrice de confusion
2.0	2.0	7	$\begin{bmatrix} 25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 18 & 4 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 & 5 \end{bmatrix}$
1.8	1.8	4	$\begin{bmatrix} 25 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$
1.0	2.0	2	$\begin{bmatrix} 25 & 0 \\ 25 & 0 \\ 0 & 25 \\ 0 & 25 \end{bmatrix}$
2.0	1.0	2	$\begin{bmatrix} 25 & 0 \\ 0 & 25 \\ 25 & 0 \\ 0 & 25 \end{bmatrix}$
1.4	1.6	4	$\begin{bmatrix} 14 & 11 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 5 & 20 \end{bmatrix}$
1.6	1.4	4	$\begin{bmatrix} 25 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 7 & 0 & 18 \end{bmatrix}$

TAB. 6.2 – Nous présentons les résultats liés aux problèmes de *Multi-Information Bottleneck* pour les données synthétisées. Nous présentons les matrices de confusion pour plusieurs clusterings obtenus avec plusieurs couples $\{\beta_1, \beta_2\}$. Nous avons 25 éléments par classe, répartis sur 4 classes. Nous remarquons par exemple que pour des paramètres de compromis égaux les deux types d'information sont pris en compte pour déterminer les clusters. D'autre part, quand $\beta_1 > \beta_2$, nous observons qu'un type d'information est privilégié pour déterminer les groupes.

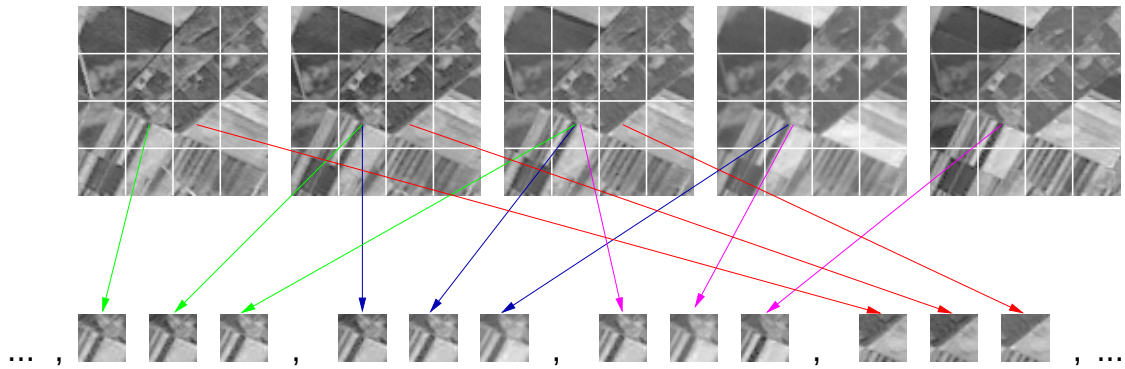


FIG. 6.5 – La séquence ci-dessus est composée de 5 images de taille 100×100. Si la taille des réalisations des champs aléatoires est $25 \times 25 \times 3$, alors nous obtenons les réalisations présentées sous la séquence. Nous notons que les pavés obtenus se recouvrent temporellement mais pas spatialement.

résumé, le principe MIB permet de faire de la fusion pondérée d'informations et de l'extraction d'information.

6.1.3 Application de la méthodologie IB aux STIS

Nous présentons quelques résultats expérimentaux obtenus pour l'extraction d'information des STIS. Nous considérons tout d'abord que la STIS est une collection de champs aléatoires indépendants et identiquement distribués. Ces champs aléatoires sont obtenus en partitionnant spatialement la série en pavés de taille $l_x \times l_y \times l_t$. D'autre part, comme la série n'est pas très étendue temporellement par rapport aux dimensions spatiales, nous acceptons un recouvrement temporel des pavés. Nous obtenons ainsi un recouvrement complet de notre STIS par ces pavés. La Figure 6.5 présente un exemple de pavage où les pavés se recouvrent temporellement. Nous présentons maintenant les résultats que nous avons obtenus sur la STIS. Premièrement, nous présentons une expérience réalisée sur une séquence de taille $100 \times 100 \times 11$ dont nous extrayons $10 \times 10 \times 7$ réalisations d'un champ aléatoire de taille $10 \times 10 \times 5$. Nous utilisons le pavage présenté dans ce paragraphe. La courbe débit-distorsion obtenue lors de cette expérimentation est tracée dans la Figure 6.6. La fonction de courbure associée à la courbe débit-distorsion y est représentée aussi. Nous voyons que le changement de comportement de cette courbe est moins franc que dans les cas présentés au §4.2.3. Néanmoins, nous voyons apparaître deux maximums locaux de la fonction de courbure. Nous sélectionnons le point de la courbe qui correspond au premier maximum local pour calculer le clustering. Avant de présenter les résultats de groupage dans l'espace des données, nous considérons la fonction d'assignement déterministe la plus proche de celle obtenue par l'algorithme IB au point critique. Pour déterminer cette fonction d'assignement déterministe ou cet index, nous considérons qu'une réalisation appartient uniquement à un cluster représenté par un des centroïdes de l'espace de reproduction. Cette procédure revient à attribuer le centroïde par MAP :

$$\tilde{z}^*(z) = \arg\max_{\tilde{z}} p(\tilde{z} \mid z) \tag{6.6}$$

$$\tilde{x}^*(x) = \arg\max_{\tilde{x}} p(\tilde{x} \mid x) \tag{6.7}$$

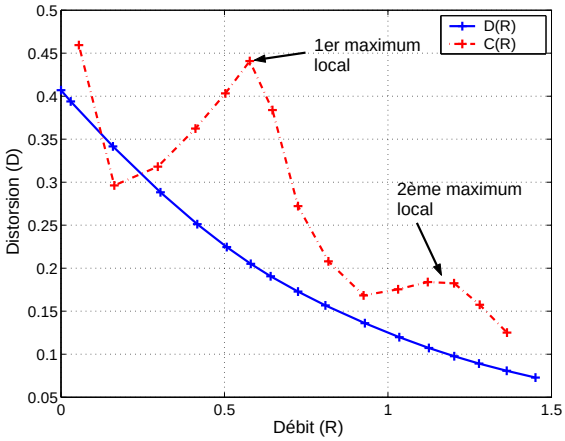


FIG. 6.6 – La courbe débit-distorsion liée à la première expérimentation est tracée. Sa fonction de courbure est représentée en parallèle où deux maximums locaux apparaissent. Ces deux maximums signifient la présence d'index optimaux.

Même si $\tilde{x}$ n'est pas calculable, nous savons comment sont grouper les réalisations. Nous représentons ce résultat de clustering pour la séquence traitée, tel que les images temporellement au centre de la réalisation sont colorées en rouge dans l'espace des données. Nous obtenons dans ce cas 3 clusters correspondant à 3 types d'événements spatio-temporels (cf. Figure 6.7). Nous avons réussit à distinguer 3 types d'événements dans cette série. L'extraction d'information sous-jacente fondée sur l'information relative aux champs aléatoires, révèle trois types de structures spatio-temporelles. Nous remarquons que le premier cluster regroupe les frontières stables et les structures linéaires. Le second cluster présente les évolutions ponctuelles. Nous pouvons voir la détection de taches d'humidité. Enfin, le dernier groupe contient les structures spatialement homogènes qui évoluent lentement dans le temps.
Nous présentons les résultats d'une seconde expérience réalisée sur une séquence de taille $200 \times 200 \times 7$. Nous utilisons la même taille pour définir les réalisations du champ aléatoire. Nous obtenons au point critique un nombre de 4 clusters représentés dans l'espace des données (cf. Figure 6.8). Nous remarquons visuellement que notre méthodologie pour l'extraction d'information réussit à discriminer les objets en fonction de l'information pertinente contenue dans les modèles. En effet, nous différencions visuellement les clusters par des considérations sur la complexité des modèles sous-jacents. Dans les deux cas traités, la quantité d'information moyenne extraite est approximativement de 15% de la quantité d'information initiale. Nous nous rapportons à l'entropie de X pour calculer ce pourcentage.
Le défaut d'une telle méthodologie est qu'elle est gourmande en temps de calcul. Sa complexité est linéaire avec le nombre de réalisations données à l'entrée de l'algorithme. Par exemple, le temps de calcul nécessaire aux deux courbes débit-distorsion est de 9 et 36 heures respectivement. Traiter des séquences de plus grande taille devient alors problématique.

6.1.4 Application de la méthodologie MIB aux STIS

Nous expérimentons le principe de *Multi-Information Bottleneck* pour l'extraction d'information des champs aléatoires de la STIS. Nous prenons les deux variables Y_1 et Y_2

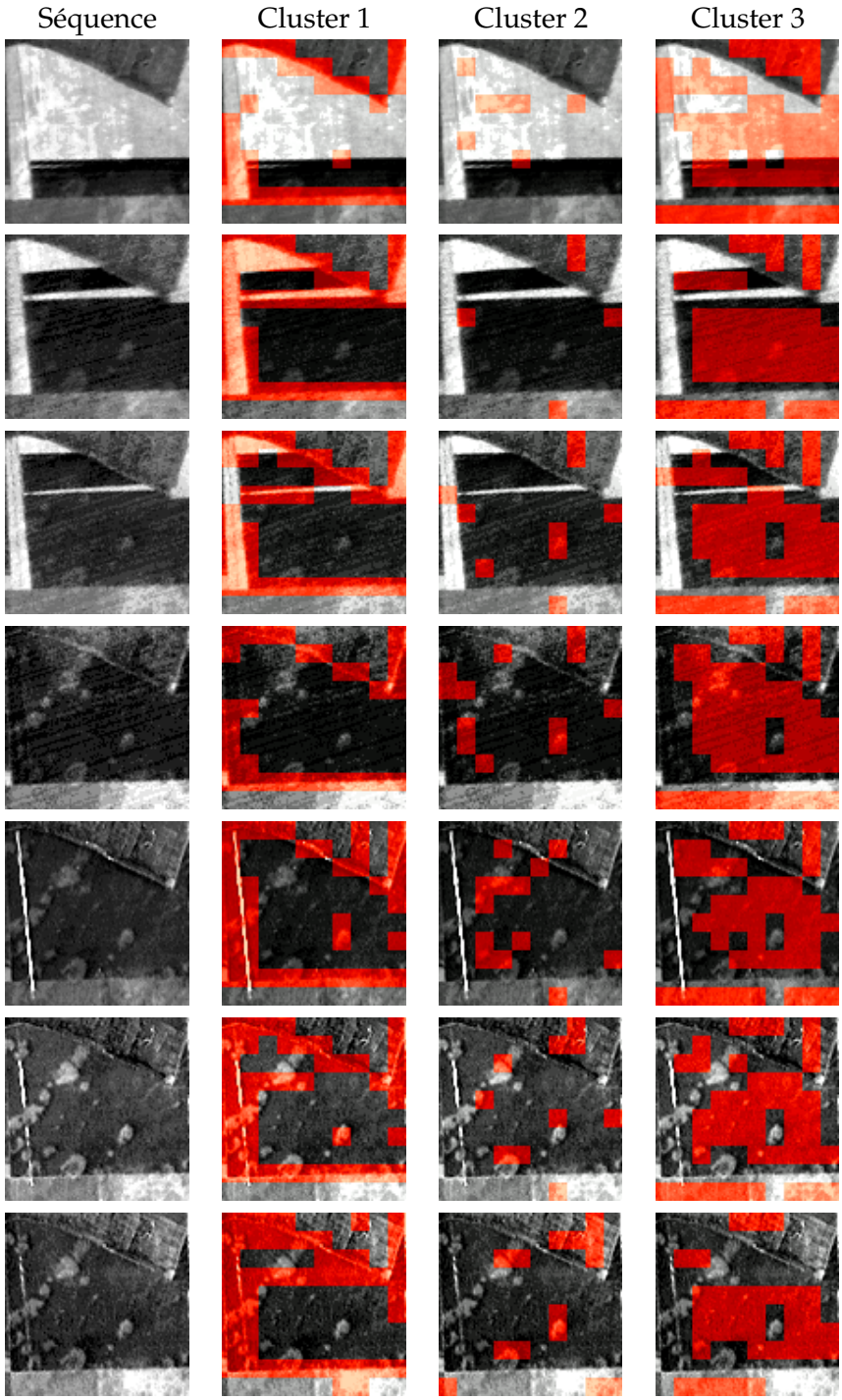


FIG. 6.7 – Nous présentons les résultats de clustering déterministe sur la série de taille $100 \times 100 \times 7$. Le nombre optimal de centroids identifiables par le critère heuristique de courbure est 3. Les 3 groupes sont présentés sur 3 colonnes qui représentent la même série. Les parties rouges montrent leur appartenance au cluster. La séquence présente la récolte d'un champ. Le premier groupe contient les frontières stables et les structures linéaires. Le second cluster présente des changements ponctuels, telle que l'apparition de tâches d'humidité. Enfin, le troisième cluster représente des changements lents de la végétation.

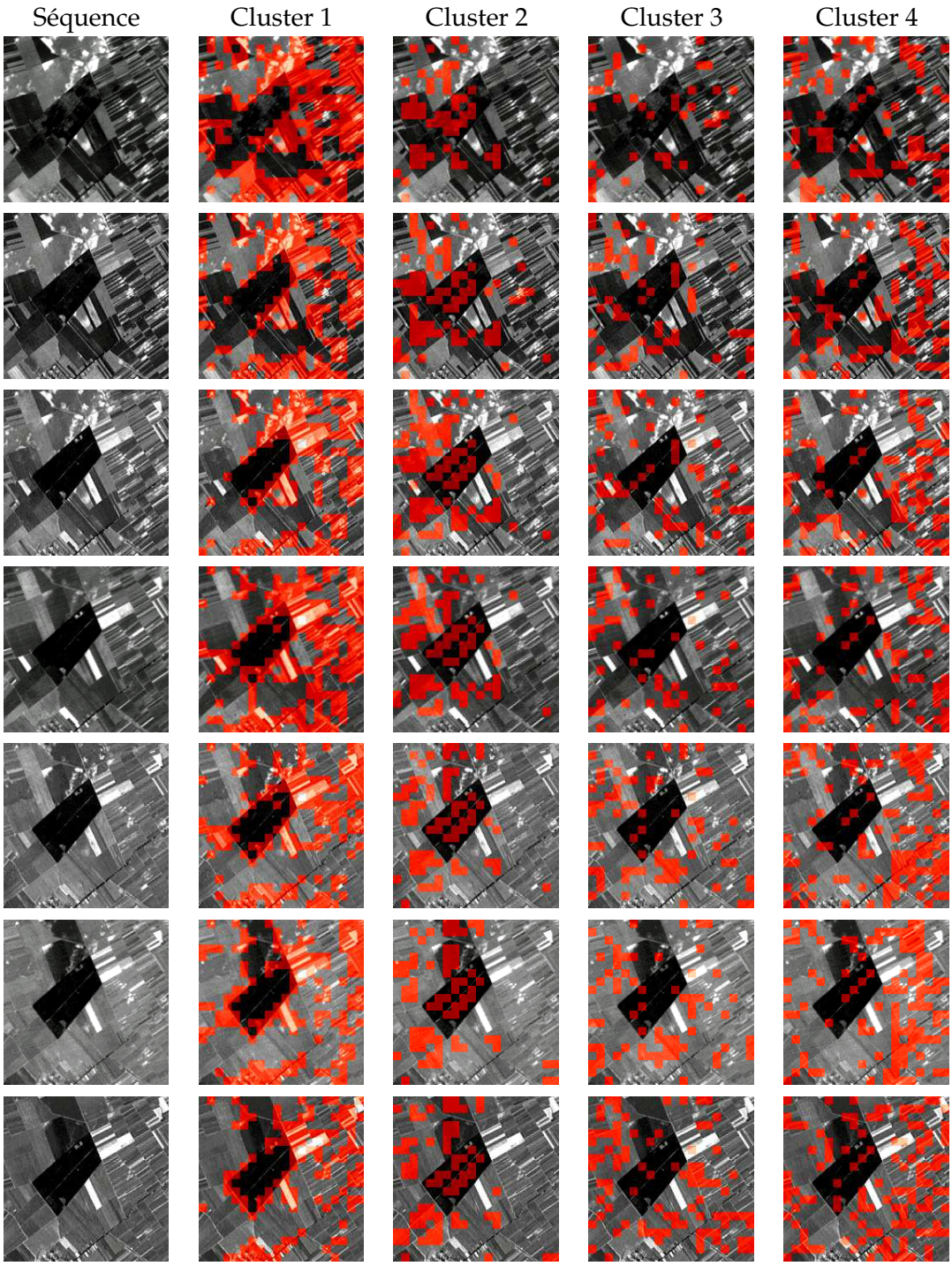


FIG. 6.8 – Nous présentons les résultats de clustering déterministe sur la série de taille $200 \times 200 \times 7$. Le nombre optimal de centroïdes identifiés par le critère heuristique de courbure est 4. Les 4 groupes sont présentés sur les 4 colonnes qui représentent la même série. Les parties rouges montrent leur appartenance au cluster. Dans le premier groupe, nous observons les frontières stables entre deux types de végétation. Le second cluster regroupe les régions qui varient lentement dans le temps. Le troisième cluster rassemble les changements ponctuels de petits objets. Finalement, le quatrième groupe contient les structures linéaires et stables comme les routes ou les talus. De plus, nous pouvons inclure dans ce groupe les champs étroits qui sont sur la partie droite des images.

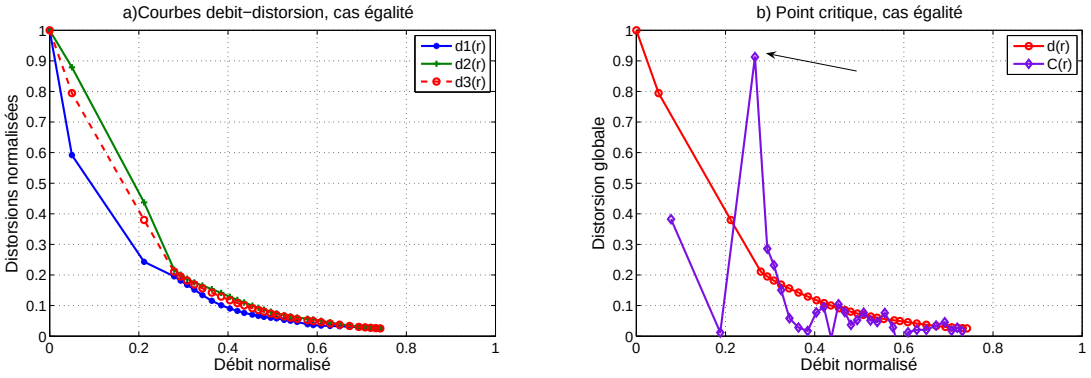


FIG. 6.9 – (a) La courbe débit-distorsion globale est tracée en rapport aux deux courbes normalisées $d^1(r)$ et $d^2(r)$. (b) La courbure de la courbe globale $d(r) = \frac{1}{2}(d^1 + d^2)(r)$ permet de trouver le point critique au maximum de la courbure. Cette fonction est notée $C(r)$. La courbure exhibe d'autres pics qui sont dus aux manques de précision. En effet, il n'y a aucun moyen de savoir si l'algorithme MIB converge vers un minimum global.

qui contiennent l'information d'importance relative à l'évolution des textures et des couleurs spectrales. Nous réalisons l'expérience sur une séquence de taille $320 \times 320 \times 5$ qui est partitionnée en 100 réalisations d'un champ de taille $32 \times 32 \times 5$. Pour notre première expérience, nous prenons la direction des paramètres de compromis telle que $[\beta_1, \beta_2] = [1/2, 1/2]$. Nous donnons autant d'importance aux deux types d'information. La courbe débit-distorsion normalisée globale est présentée dans la Figure 6.9. Nous trouvons le point critique, et calculons le groupage flou correspondant. Il y a 15 clusters à cet optimum. La Figure 6.10 présente le résultat de ce clustering dans l'espace des données.

D'autre part, nous rééditons l'expérience en prenant comme vecteur de compromis $[0.3, 0.7]$. Ainsi, nous privilégions l'information relative à l'évolution des couleurs. Nous donnons la courbe débit-distorsion normalisée globale dans la Figure 6.11. Le point critique correspond à 18 centroïdes identifiables. Nous présentons quelques clusters dans l'espace des données dans la Figure 6.12. Enfin, dans le cas ou le vecteur de compromis est $[0.8, 0.2]$, l'information texturale est privilégiée. Les Figures 6.13 et 6.14 montrent les résultats. Nous obtenons dans ce dernier cas 6 clusters.

Nous donnons une brève comparaison des 3 résultats de clustering. Premièrement, nous notons que le nombre de clusters identifiables obtenus dans le premier cas se situe entre les deux autres cas extrêmes. Il paraît logique qu'il en soit ainsi puisque nous passons continument du deuxième cas vers le premier pour finir avec le troisième. Les résultats visuels obtenus sont satisfaisant et trouvent une explication sémantique. Nous préconisons de privilégier autant les deux types d'information et ainsi obtenir les résultats de clustering de la Figure 6.9. Ainsi, l'information extraite satisfait l'intérprétation visuelle donnée. Elle est plus en harmonie avec ce que l'homme observe dans la STIS. Nous notons que dans le premier cas, le pourcentage d'information extraite représente 25% de l'information initiale. En comparaison à la méthodologie fondée sur le principe IB, le pourcentage d'information extraite est plus important puisque nous avons considéré l'information de couleur.

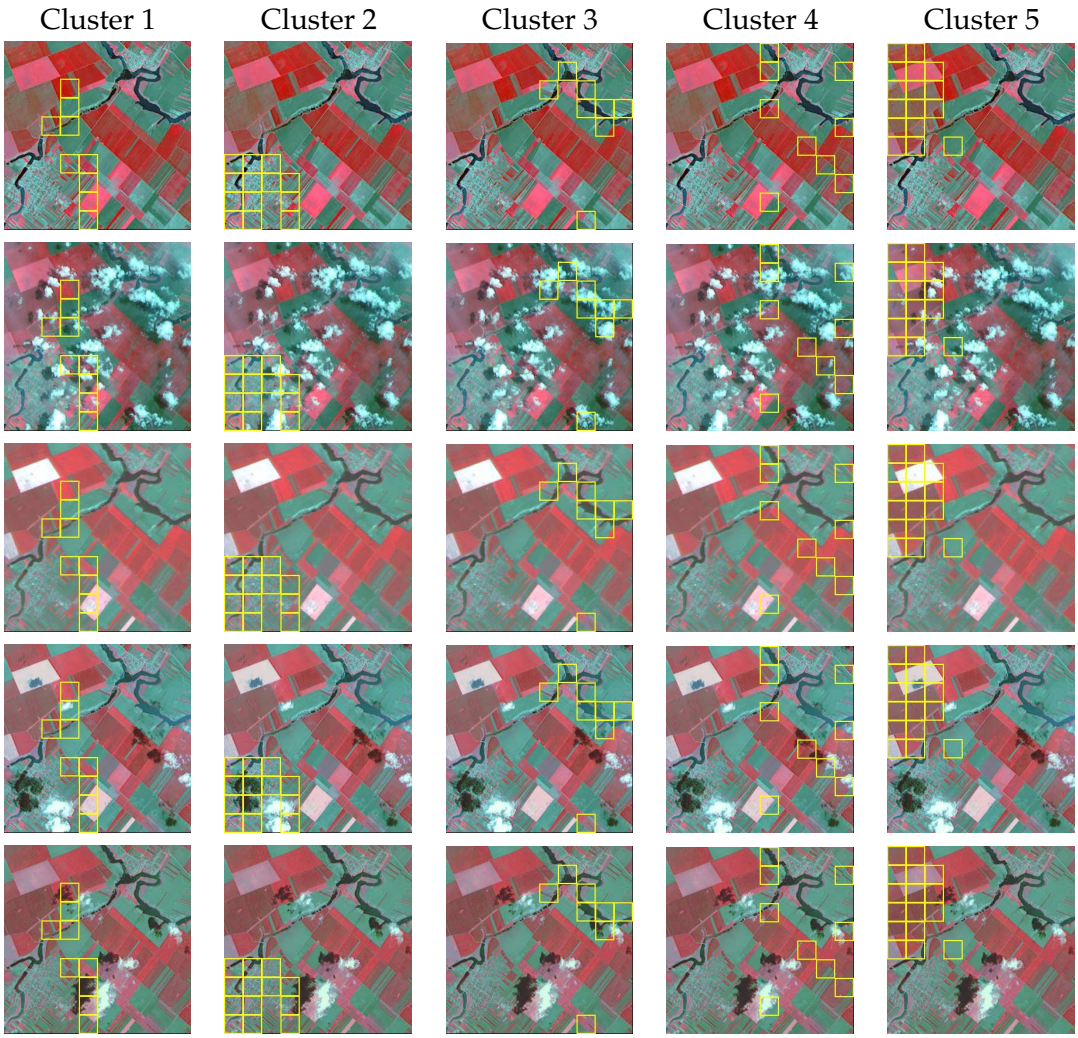


FIG. 6.10 – Cette figure présente 5 des 15 clusters dans l'espace des données. Dans ce clustering les deux types d'information considérés ont la même importance. Nous remarquons que le cluster 1 regoupe les structures qui subissent une occlusion de nuages à l'instant 2 et subissent une occlusion par une ombre au dernier instant. La coloration est divisée équitablement en vert et rouge. Nous remarquons que ces événements présentent de fines structures linéaires appraissent. Le cluster 2 regroupe les structures spatio-temporelles liées à la ville dont les couleurs mixtes varient peu au cours du temps. Le cluster 3 représente les rivières qui sont couvertes par des nuages. Enfin le cluster 5 regroupe les parcelles en pleine floraison. Les événements présentent une coloration rouge, tandis que les textures sont spatialement homogénes.

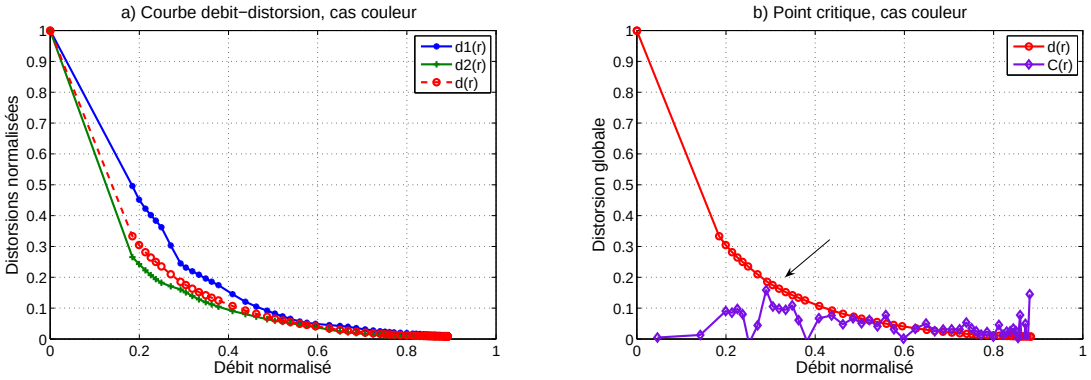


FIG. 6.11 – L'information de couleur est privilégiée en prenant $\vec{\beta} = [0.3, 0.7]$. (a) La courbe débit-distorsion globale est tracée en rapport aux deux courbes normalisées $d^1(r)$ et $d^2(r)$. (b) La courbure de la courbe globale $d(r)$ permet de trouver le point critique au maximum de la courbure.

6.1.5 Analyse des expériences réalisées sur la STIS

Nous nous rapportons dans un premier temps aux résultats présentés dans les Figures 6.7,6.8. Ces groupages sont peu satisfaisants visuellement. Même, si certains types de structures spatio-temporelles sont bien regroupés, l'interprétation possible des groupes est fondée sur des critères très simples tels les changements abrupts ou les structures linéaires. Néanmoins, ces expériences montrent la pertinence des champs aléatoires de Gauss-Markov ou autobinomiaux pour extraire l'information des STIS, en incorporant une sélection de modèle. C'est cette observation qui nous a amené à conserver ce type de modélisation dans la construction des prédicteurs du codeur sans perte.
Dans un second temps, nous nous rapportons aux expériences réalisées sur la STIS avec le principe MIB. Nous avons vu une amélioration des résultats en ce qui concerne le groupage par couleur. Mêmes si les résultats ne sont pas satisfaisant, ils nous ont permis de démontrer l'existence de deux types d'information indépendants dans la STIS. Aussi, nous nous rendons compte que l'information de couleur apporte beaucoup dans l'indexation d'événements spatio-temporels.
En résumé, ces méthodologies fondées sur le principe IB nous ont permis d'étudier la modélisation de la STIS et de comprendre les types d'informations présents. Le gros désavantage de ces méthodes réside dans leur complexité calculatoire. En l'occurrence, il est possible de traiter seulement des petites sous-séries séparément au lieu de la STIS complète.

6.2 Validation de la méthodologie fondée sur les complexités

Dans cette partie, nous souhaitons valider notre méthode de compression/indexation sur des données synthétiques. Nous souhaitons montrer que l'algorithme permet d'inférer une classification non-supervisée correcte et permet de déterminer le nombre de clusters optimal. Ensuite, nous comparons cette dernière méthode à la méthode fondée sur le principe IB.
Nous appliquons la méthodologie fondée sur les complexités aux STIS. Dans un premier temps, nous validons le codeur par les taux de compression obtenus. Ensuite, nous décrivons une procédure pour diminuer la complexité de notre algorithme dans le but de

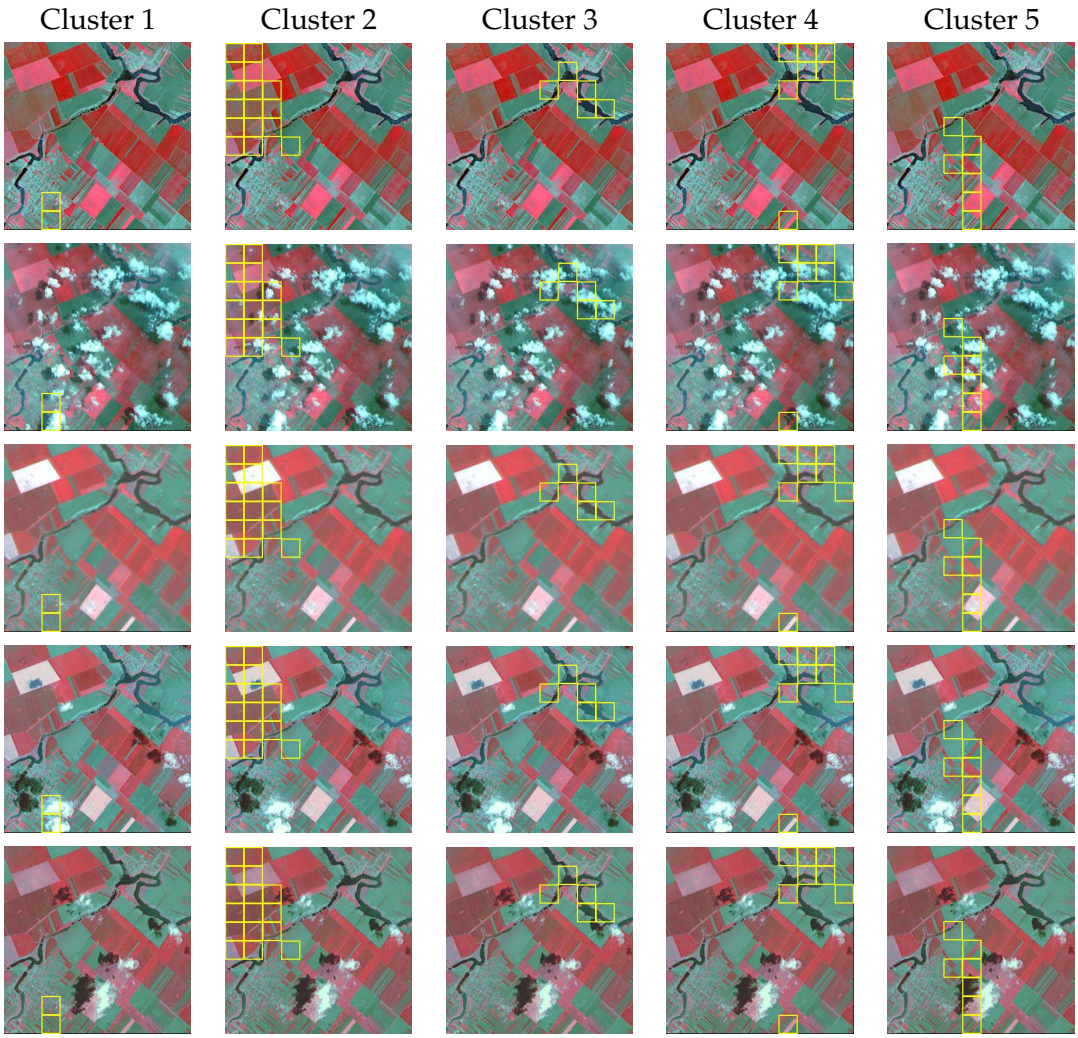


FIG. 6.12 – Cette figure représente 5 clusters des 18 clusters obtenus au point critique. Dans ce cas l'information liée aux couleurs est privilégiée. Nous voyons par exemple que le cluster 1 regoupe deux structure spatio-temporelles qui correspond à une double occlusions par les nuages. Le cluster 3 présente la riviére entourée de vert qui est cachée par des nuages au deuxième instant. Enfin, le cluster 4 présente les structures autour des villes qui sont couvertes par la brume au premier instant.

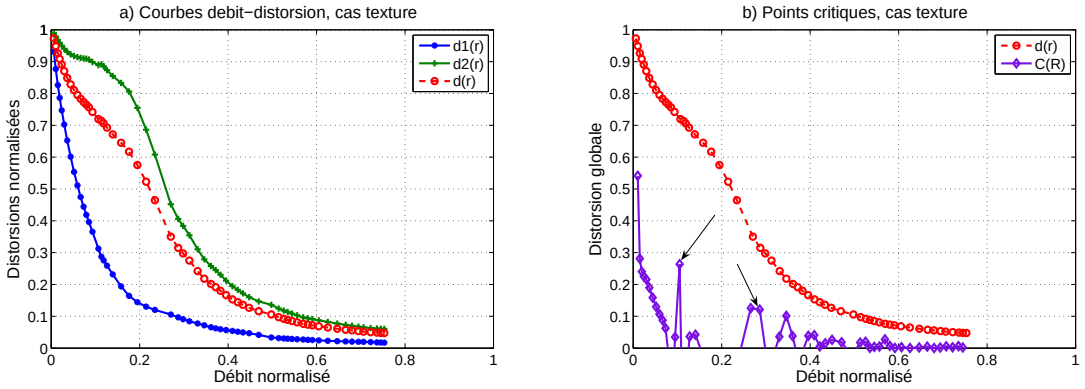


FIG. 6.13 – L'information de texture est privilégiée en prenant $\vec{\beta} = [0.8, 0.2]$. (a) La courbe débit-distorsion globale est tracée en rapport aux deux courbes normalisées $d^1(r)$ et $d^2(r)$. (b) La courbure de la courbe globale $d(r)$ permet de trouver le point critique au maximum de la courbure.

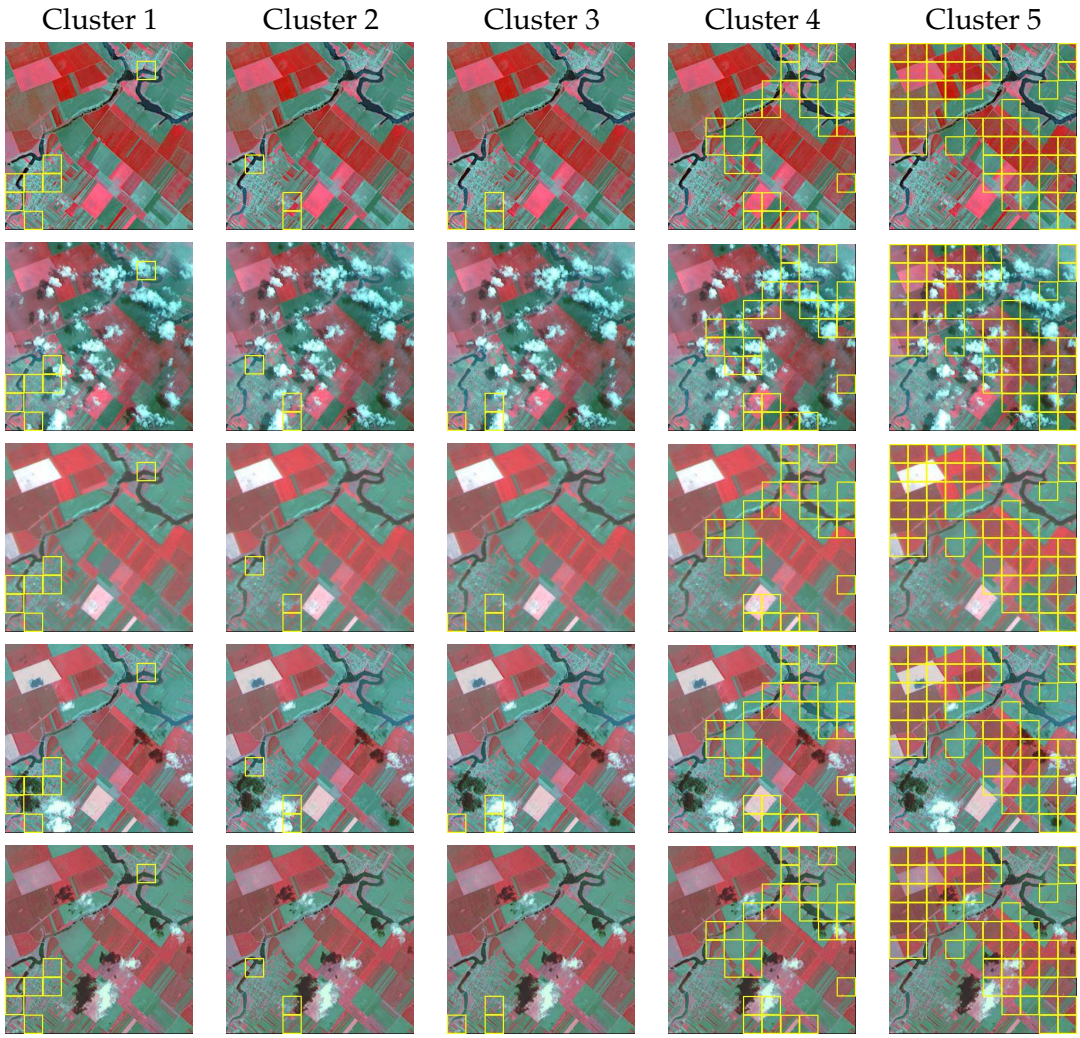


FIG. 6.14 – Cette figure présente 5 clusters parmi les 6 obtenus. Dans ce cas, l'information texturale est privilégiée. Nous remarquons que les structures spatio-temporelles sont homogénes en termes d'évolution de texture. Par exemple, le cluster 5 regroupe les régions spatialement étendues qui évoluent peu temporellement.

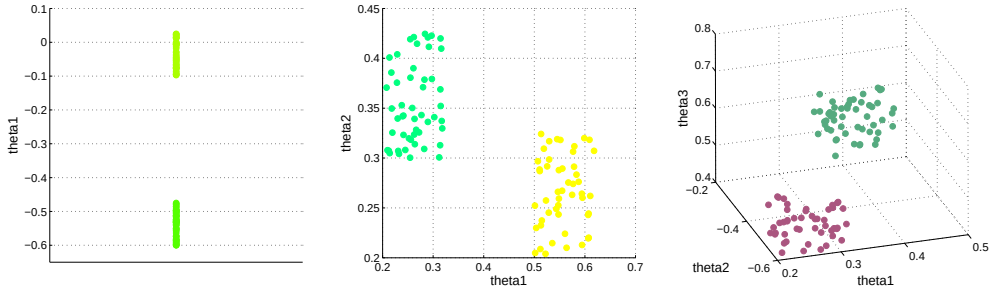


FIG. 6.15 – Les différents paramètres des modèles autorégressifs sont représentés dans leur espace respectif. Ces paramètres ont été générées et permettent de synthétiser les signaux aléatoires correspondants. Nous distinguons 2 classes par espace, ce qui nous donne 6 classes au total. Le code de couleur indique l'appartenance à une classe. Les classes 1 à 6 sont, dans l'ordre, de couleur vert clair, vert, vert brillant, jaune, vert foncé et mauve.

traiter la STIS complète. Enfin, nous donnons un résultat de clustering, avant de valider notre extraction de la STIS ADAM en intégrant un système de recherche par exemple.

6.2.1 Validation sur des données synthétiques

Pour évaluer notre méthodologie emprunt du principe LDM, nous nous plaçons dans le cas de processus stochastiques. Nous considérons encore des signaux monodimensionnels autorégressifs comme au §6.1.1. Cependant, cette fois ci nous ne considérons pas une distribution uniforme des paramètres. Nous prenons les modèles autorégressifs des trois premiers ordres, dont les espaces des paramètres sont inclus respectivement dans $\mathbb{R}$,$\mathbb{R}^2$ et $\mathbb{R}^3$. Dans chaque espace nous considérons que les paramètres sont distribués selon un mélange de deux gaussiennes. Par conséquent, nous nous trouvons avec six classes structurées en deux classes par espace. Ces paramètres sont représentés dans la Figure 6.15. Ces paramètres permettent de synthétiser les signaux aléatoires en filtrant des bruits blancs gaussiens de variance unitaire. Pour tester notre méthode fondée sur le principe LDM, nous réalisons une inférence en deux niveaux des paramètres pour chaque signal. Tout d'abord les paramètres sont estimées par Maximum de Vraisemblance pour chaque type de modèle et pour chaque objet x. Ensuite, le meilleur modèle est sélectionné. Enfin, les paramètres correspondant au modèle sélectionné constituent le modèle $\mathcal{M}_x$ de notre objet x. Cette étape d'estimation correspond à un codage en deux parties dans un cadre statistique (cf. §3.3.3). Les paramètres estimés sont présentés dans la Figure 6.16 en fonction de leur appartenance aux six classes.
Nous avons défini la DNSC pour les processus stochastiques dans l'Eq. 5.18. Par conséquent, il est possible d'appliquer un codage conjoint des différents processus pour construire un clustering des signaux. L'algorithme de clustering (cf. §5.2.3) nous donne 7 clusters et un index dont la matrice de confusion avec les classes initiales est donnée dans le Tableau 6.3. Nous remarquons que l'estimation du nombre de clusters est très proche du nombre de classes, même si celles-ci n'apparaissent pas de manière franche dans les espaces des paramètres estimés. La matrice de confusion montre que l'algorithme réussit à récupérer les caractéristiques des signaux puisqu'il n'y a pas beaucoup de confusion. La Figure 6.17 montre les résultats de clustering dans l'espace des paramètres estimés. Nous remarquons que cette classification non supervisée est très proche de la classification initiale. La procédure de minimisation fondée sur la DNSC a réussit à trouver les

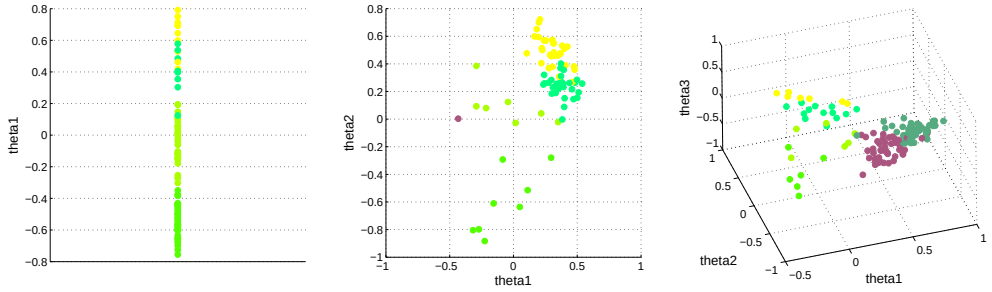


FIG. 6.16 – Les paramètres sont estimés en deux niveaux d'inférence à partir des signaux synthétiques. Nous présentons leur appartenance aux classes initiales par les mêmes codes de couleurs présentés dans la Figure 6.15. Nous remarquons, que les classes initiales ne sont plus aussi bien discriminées après l'estimation. D'autre, part nous remarquons qu'une classe peut s'étaler d'un ordre à l'autre.

classes/ clusters	1	2	3	4	5	6	7
1	49	6					
2		37		3		1	4
3			30	19			
4			3	48			
5					50		2
6					13	29	6

TAB. 6.3 – Ce tableau présente la matrice de confusion entre l'index originale des classes et l'index obtenu par codage et clustering. Nous notons que le nombre de clusters est proche du nombre de classes. D'autre part, nous observons que la matrice est peu confuse. Les seules confusions apparaissent au sein du même ordre de modèles.

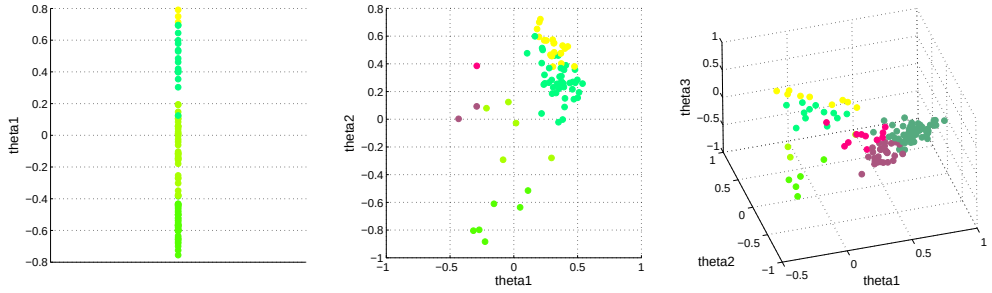


FIG. 6.17 – Cette figure présente les résultats de clustering dans l'union des espaces des paramètres estimés. Nous observons, que les groupes principaux sont en adéquation avec les classes initiales présentées dans la Figure 6.16.

ressemblances au travers des trois espaces. Par conséquent, ces résultats montrent l'efficacité de notre méthodologie pour extraire l'information pertinente contenue dans les données en exploitant le codage en deux parties.

6.2.2 Comparaison aux méthodes de références

Nous prenons le même type de données synthétiques présentées au paragraphe précédent. Ce sont des signaux autorégressifs des trois premiers ordres où 6 classes sont générées aléatoirement à plusieurs reprises pour constituer plusieurs bases de tests. Nous voulons comparer les performances de notre méthodologie aux méthodes fondées sur AutoClass et k-moyennes. Nous utiliserons les mesures d'entropie et de pureté pour valider notre méthode.
Tout d'abord, nous décrivons les classifications non supervisées et fondées sur les algorithmes Autoclass et k-moyennes. L'évidence des modèles autorégressifs est calculée indépendamment pour chaque signal de l'ensemble synthétique. Par un critère de Maximum de Vraisemblance, nous sommes capables de déterminer le meilleur modèle de l'ensemble des signaux. Ce meilleur modèle est directement lié à l'ordre et l'espace des paramètres. Une fois l'espace des paramètres fixé, nous estimons les paramètres modélisant aux mieux chaque signal autorégressif. Nous pouvons observer une confusion des classes dans cet espace. Ensuite, nous appliquons l'algorithme AutoClass aux primitives extraites. Cet algorithme permet de déterminer un nombre optimal de clusters et un index des signaux. D'autre part, nous appliquons l'algorithme des k-moyennes sur les primitives en fixant le nombre de clusters à être égal au nombre de classes. Par ces deux comparaisons, nous voulons montrer d'un côté que notre algorithme donne une bonne estimation du nombre de classes. D'un autre côté, nous voulons montrer qu'il est plus général puisqu'il permet de travail dans des espaces à multiples dimensions.
Pour évaluer les différents index obtenus, nous les comparons à la classification de référence en s'appuyant sur les mesures d'entropie E et de pureté P définies par :

$$E = \sum_{r=1}^{k} \frac{n_r}{n} \frac{1}{\log q} \sum_{i=1}^{q} \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r} \tag{6.8}$$

$$P = \frac{1}{n} \sum_{r=1}^{k} \max_i n_r^i \tag{6.9}$$

	DNSC + LDM			AutoClass			k-moyennes		
	E	P	k	E	P	k	E	P	k
moyenne	0.40	0.70	6.5	0.55	0.53	3.4	0.47	0.64	6
variance	0.09	0.10	3.86	0.18	0.17	1.54	0.10	0.10	0

TAB. 6.4 – Ce tableau présente les mesures de qualité des clustering obtenus pour les 3 méthodes. Les moyennes et les variances des mesures d'entropie et de pureté sont présentées pour chaque méthode. De plus la moyenne et la variance du nombre optimal de clusters k sont présentée pour les clusterings fondés sur les complexités et AutoClass .

où n est le nombre d'objets, n_r est le nombre d'éléments dans le cluster r, n_r^i est le nombre d'objets de la classe i appartenant au cluster r, k est le nombre de clusters et q est le nombre de classes. Plus l'entropie est basse et la pureté élevée, meilleur est le clustering en comparaison aux classes prédéterminées. Ces quantités sont normalisées. Nous appliquons les 3 méthodes sur 30 ensembles de 300 signaux synthétiques réparties sur 6 classes. Nous présentons dans le Tableau 6.4, les moyennes et variances des différentes mesures de qualité obtenues pour chaque ensemble. Dans un premier temps, nous notons que notre algorithme donne une estimation du nombre de clusters moins biaisé qu' AutoClass. Cependant, la variance de nombre de clusters dans le cas de notre algorithme s'en trouve augmenté. L'avantage de notre méthode vient du fait que nous travaillons dans une union d'espaces, alors que AutoClass travaille dans un espace de dimension fixée. Ainsi, la confusion provoquée dans cet espace réduit le nombre de clusters estimé par AutoClass. D'autre part, nous notons que les résultats en termes d'entropie et de pureté sont meilleurs avec l'algorithme fondé sur les complexités. En définitif, ces expériences valident notre algorithme sur des données synthétiques complexes. Nous notons toutefois que les résultats de clustering sont loin d'être bons. Cela est dû essentiellement à la confusion qui est réalisée lors de la sélection de modèle dans l'inférence en deux niveaux.

6.2.3 Comparaison à la méthodologie fondée sur le principe IB

Nous comparons la méthodologie fondée sur le principe IB à celle fondée sur les complexités. Nous prenons le même cadre expérimental décrit au §6.2.2. Cependant celui-ci diffère sur un point. Au lieu de considérer autant de bruits blancs gaussiens que d'objets, nous prenons un unique signal gaussien auquel nous appliquons les modèles autorégressifs. Nous obtenons ainsi autant de signaux que de vecteurs de paramètres générés. Cet ensemble de signaux est toujours classifié en 6 catégories qui ont permis de générer les paramètres.
Dans le but d'appliquer le principe IB, nous définissons la variable aléatoire $\mathcal{M}$ qui prend ses valeurs dans l'ensemble des modèles autorégressifs causaux des 4 premiers ordres. Nous calculons la courbe débit-distorsion dans ce contexte, avant de déterminer le paramètre de compromis optimal qui déterminera le nombre de clusters.
Nous appliquons les deux méthodologies sur plusieurs ensembles d'objets dont les paramètres sont générés aléatoirement. Nous évaluons les clusterings en termes d'entropie, de pureté et du nombre de clusters. Les résultats sont présentés dans le Tableau 6.5. Nous remarquons que la méthodologie fondée sur le principe IB est plus efficace au regard des mesures de qualité moyenne. Cependant, nous notons une faiblesse sur l'estimation du nombre de classes en moyenne. D'autre part, nous remarquons que la méthodologie fondée sur le principe IB donne des résultats très variables en comparaison

	Information Bottleneck			DNSC + LDM		
	E	P	k	E	P	k
moyenne	0.22	0.78	11.3	0.33	0.74	5.8
variance	0.26	0.22	6.1	0.18	0.08	1.5

TAB. 6.5 – Ce tableau présente les mesures de qualité des clustering obtenus pour les 2 méthodes introduites dans notre travail. Les moyennes et les variances des mesures d'entropie et de pureté sont présentées pour chaque méthode. De plus la moyenne et la variance du nombre optimal de clusters k sont présentées.

bande	erreur (bpp)	arbre (bpp)	prédicteur (bpp)	total (bpp)
1	2.360	0.040	0.002	2.402
2	2.898	0.043	0.002	2.943
3	3.921	0.043	0.002	3.966

TAB. 6.6 – Les débit moyens de codage sans perte en bpp de la STIS ADAM sont présentés. Le débit moyen nécessaire au codage des modèles est très faible en comparaison du débit total.

de la méthodologie fondée sur les complexités. Il apparaît que cette dernière méthodologie est plus consistante que la méthodologie IB pour extraire l'information, en apportant une bonne estimation du nombre de classes et en produisant des index satisfaisants.

6.2.4 Résultats expérimentaux obtenus sur les STIS

Dans cette section, nous évaluons les performances de notre méthodologie sur la STIS ADAM. D'abord, nous présentons les performances de notre codeur sans perte (cf. §5.3). Ensuite, nous donnons une procédure pour simplifier notre algorithme de codage conjoint dans le but de l'appliquer à la STIS. Enfin, nous présentons quelques groupes obtenus par l'algorithme dans l'espace des données.

6.2.4.1 Taux de compression

Nous expérimentons notre codeur sans perte sur la STIS ADAM et comparons les taux de compression à l'état de l'art de la compression réalisé sur la STIS.
Nous partitionnons la STIS en blocs spatio-temporels qui ne se chevauchent pas. Nous prenons arbitrairement la taille des blocs à être $32 \times 32 \times 5$, et nous effectuons l'apprentissage des prédicteurs pour chaque bande spectrale. En fait, le choix de la taille s'est fait de telle manière que le signal soit spatialement stationnaire et que le nombre de réalisation d'erreur soit significativement plus grand que le nombre de contextes. Dans notre cas, nous codons $32 \times 32 \times (5-1) = 4096$ échantillons d'erreurs par bloc. Nous expérimentons notre codeur séparément sur les 3 bandes spectrales avec 25000 blocs par série. Nous obtenons les résultats de compression présentés dans le Tableau 6.6. Les résultats sont donnés en bits par pixel pour chaque bande. Ces résultats ne prennent pas en compte la longueur de codage de la première image, puisque nous nous intéressons au codage des images suivantes. Nous comparons nos résultats de compression sans perte à plusieurs méthodes de compression quasi sans perte appliquées à la STIS (Gueguen et al., 2005). Ces méthodes de compression sont fondées sur les décompositions en ondelettes, la TCD ou la transformée de Kahrunen-Loéve pour décorreler au maximum les données.

De plus les résultats donnés sont quasi sans perte, c'est-à-dire que des petites erreurs son tolérées. Les meilleurs débits obtenus en compressant séparément les trois bandes 1,2 et 3 sont 2.46 bpp, 2.84 bpp et 4.42 bpp. Nous constatons que notre codeur donne de meilleures performances, ce qui signifie que nous obtenons une meilleure modélisation des STIS. Tandis que les méthodes de compression fondées sur les transformées sont globales, notre codeur n'exploite pas les redondances entre blocs. Ainsi nous espérons réduire le débit moyen en appliquant notre algorithme d'extraction d'information et de codage conjoint.
Finalement, nous observons que la quantité d'information extraite représente en moyenne 1% de l'information totale.

6.2.4.2 Codage d'une grande base de données

Nous appliquons notre algorithme d'extraction d'information (cf. §5.2.1) à l'ensemble des blocs ou l'ensemble des événements spatio-temporels. Nous traitons les séries monospectrales séparément. Nous rappelons que notre algorithme a une complexité en $O(n^2)$ où n est le nombre de blocs. Dans le cas des STIS, ce nombre est trop important pour pouvoir exécuter notre algorithme. Pour résoudre le problème, nous proposons une procédure sous optimale de clustering qui permet de réduire la complexité.
Nous décrivons brièvement cette procédure qui consiste à séparer les n objets initiaux en sous-groupes de taille p. Ensuite, nous appliquons la méthodologie de codage à chaque ensemble pour obtenir leurs modèles et leur index. Soit n_1 le nombre total des centroïdes résultants de ces groupages. Nous prenons ces n_1 centroïdes et nous réitérons le processus. L'algorithme itératif s'arrête quand le nombre de centroïdes est proche du nombre d'objets avant les groupages. Quand ce critère est vérifié, l'information mutuelle inter-objets devient très petite et insignifiante puisque le groupage n'a pas réussit à exploiter les redondances entre objets. A chaque itération, nous conservons les index de chaque groupe qui permettent de créer un index hiérarchique dont la hauteur de l'arbre associé est égale au nombre d'itérations. Cette procédure sous optimale permet de réduire significativement la complexité de l'algorithme de codage. Admettons que les objets sont réduits dans une proportion κ fixe à chaque itération. C'est une approximation de notre algorithme, puisque si ce coefficient de proportionnalité est fixe, l'algorithme ne converge pas. Néanmoins, la complexité de la première itération est $\frac{n}{p}O(p^2)$, celle de la deuxième est $\frac{\kappa n}{p}O(p^2)$ et ainsi de suite. Alors la complexité totale devient quand les itérations se suivent à l'infini :

$$\sum_{i=0}^{\infty} \frac{\kappa^i n}{p} O(p^2) = O(np) \sum_{i=0}^{\infty} \kappa^i \tag{6.10}$$

Etant donné que $\kappa < 1$, le terme de somme est un constante et l'algorithme itératif à une complexité de calcul de $O(np)$ au lieu de $O(n^2)$. En réduisant suffisamment le nombre p, la procédure devient réalisable sur des grands ensembles d'objets.
Pour exploiter au mieux les redondances inter-objets et pour améliorer la procédure précédente, nous préconisons deux critères très simples pour former les groupes de taille p. Par exemple, il est possible de grouper a priori les objets par leurs complexités ou leurs longueurs de code. En effet, il est difficilement envisageable que deux objets de complexités complètement opposées soient fortement similaires. D'autre part, il est possible de lire les blocs en suivant un courbe de Peano pour conserver la proximité spatio-temporelle lors de la formation des sous-groupes de taille p. Ce dernier critère est exclusivement dédié aux images ou STIS codées par blocs.

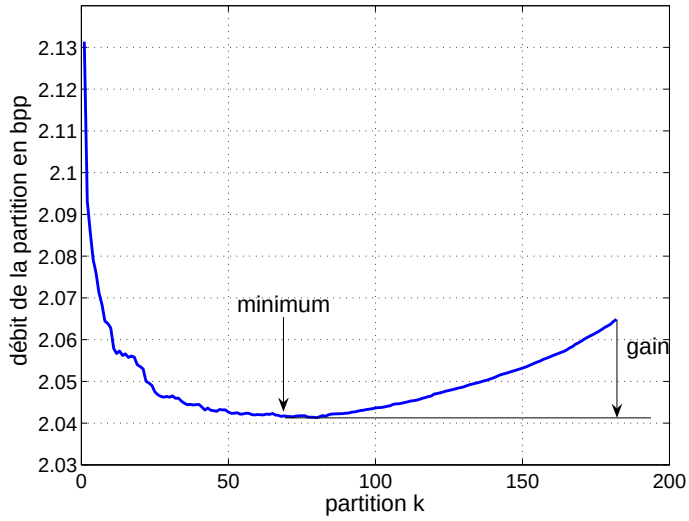


FIG. 6.18 – Cette courbe montre les variations de la longueur de codage moyenne $L(\mathcal{P}_k)$ en fonction du nombre de clusters. Le débit est calculé en bpp. On distingue nettement le minimum correspondant au critère LDM et le gain de codage réalisé.

p	30	70	100	200	400
gain en bpp	0.0081	0.0100	0.0111	0.0131	0.0133

TAB. 6.7 – Ce tableau présente l'évolution du gain de codage exprimé en bpp par rapport au nombre p d'objets jointement codés. Il est évident que plus p est grand, meilleur est le gain de codage puisque plus de redondances sont éliminées. En observant la courbe de gain, nous remarquons que la croissance est logarithmique avec le nombre p. Cette modélisation sommaire permet d'extrapoler le gain de codage maximum réalisable avec $p = n$ sur la STIS, qui serait de 0.0239 bpp.

6.2.4.3 Application aux STIS

Dans un premier temps, nous donnons le gain de codage réalisé sur une sous-série, dont le nombre d'objets spatio-temporels s'élève à 186. Nous donnons dans la Figure 6.18, l'évolution de la longueur de code globale en fonction des partitions $\mathcal{P}_k$. D'après la méthodologie décrite, nous sélectionnons le minimum de la courbe. Si les n objets sont codés indépendamment, cela revient à considérer la partition $\mathcal{P}_n$. Ainsi, le gain de codage se lit comme la différence entre le minimum et le bout de la courbe. Nous observons dans notre exemple de la Figure 6.18 que nous obtenons un gain moyen de 0.02 bpp. Pour appréhender l'influence du paramètre p, nous appliquons notre procédure d'optimisation rapide sur une même sous-série monospectrale. Le Tableau 6.7 montre les gains moyens en fonction du nombre d'objets p. Nous remarquons que ce gain augmente avec p puisque plus de redondances sont prises en compte à la fois et donc éliminées. En conséquence, il y a un compromis à réaliser sur la valeur de p pour que le gain soit le plus grand possible tout en limitant la complexité de l'algorithme d'indexation rapide.
Dans un second temps nous présentons des résultats visuels de groupage sur une sous séquence extraite de la STIS, dans la Figure 6.19. Par exemple nous remarquons le groupage d'événements qui présentent une occlusion par un nuage. Finalement, l'indexation de cette séquence ou de cet ensemble d'événements prend approximativement 0.5h. Cette complexité calculatoire est largement en dessous de celle obtenue avec la méthodologie

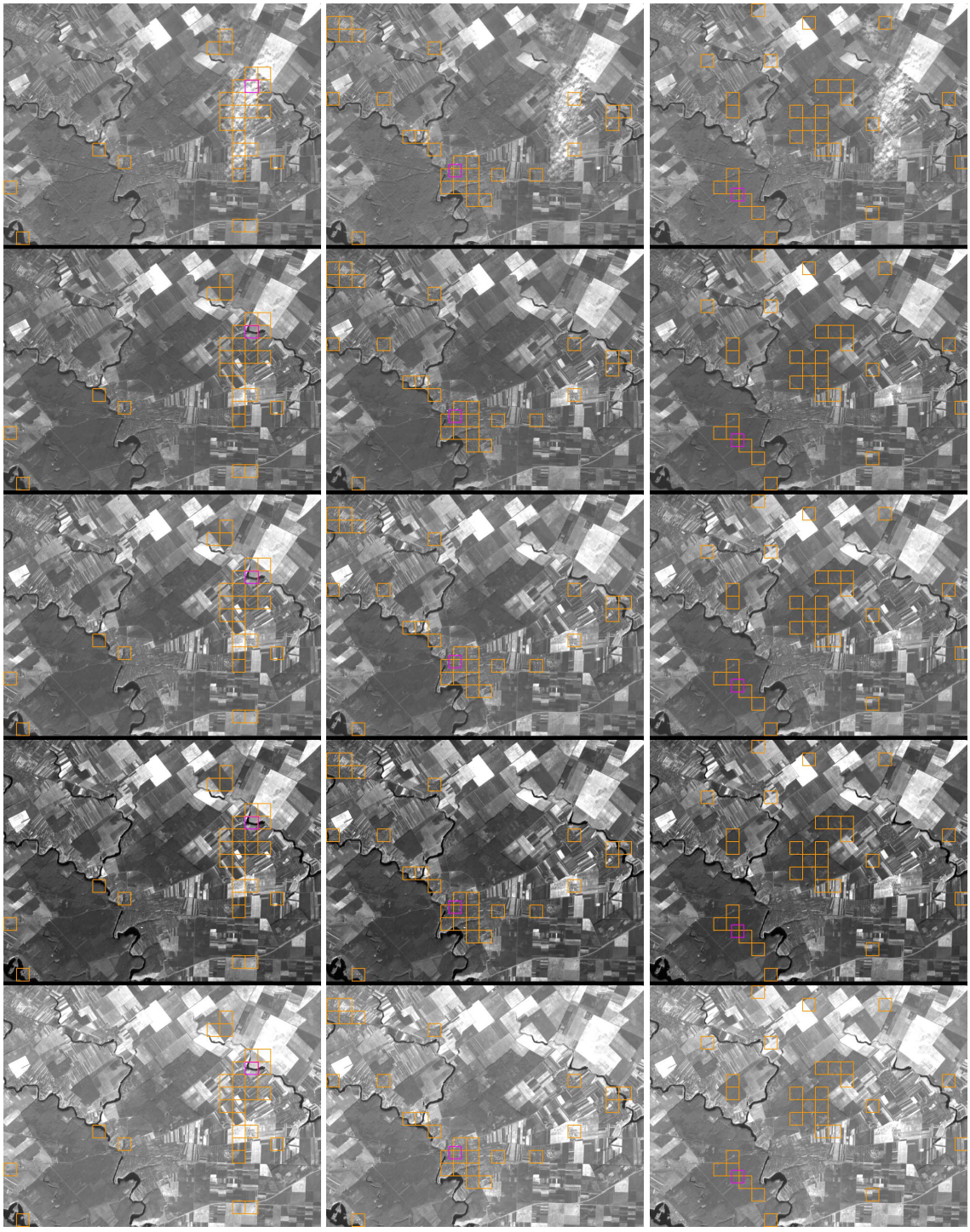


FIG. 6.19 – Trois groupes de structures spatio-temporelles sont présentés, après avoir clustériser une série de taille $800 \times 600 \times 5$ divisée en pavés de taille $32 \times 32 \times 5$. L'ensemble d'événements est constitué de 475 objets. Le premier groupe contient les occlusions par des nuages. La seconde série regroupe les villes stables. Finalement, le troisième groupe présente les zones de champs qui ne changent pas.

FIG. 6.20 – La série présentée se lit de gauche à droite et de haut en bas. Cette figure présente les résultats de requête par l'exemple qui est délimité spatio-temporellement par la suite de rectangles magentas. Les réponses sont délimitées par les suites des carrés oranges. Avec l'exemple donné, nous voulions retrouver des champs très fins, quasi linéaires, qui sont traversés par un nuage brumeux. Sur toute la séquence, les résultats les plus pertinents sont présentées dans cette sous série de taille $600 \times 600 \times 7$ dont l'exemple a était tiré.

IB.

6.2.5 Analyse de l'extraction d'information réalisée sur la STIS

Pour analyser les résultats de l'extraction d'information réalisée sur la STIS, nous mettons en place un moteur de recherche par le contenu. Celui-ci prend en entrée un exemple d'événement spatio-temporel de dimension équivalente aux structures spatio-temporelles indexées. Ensuite, cette exemple est comparé à tous les modèles qui constituent l'index (cf. Figure 5.1) en s'appuyant sur la mesure de similarité décrite dans l'Eq. 5.23. Nous conservons un nombre fixe des modèles les plus proches. Ensuite, nous décodons et présentons les structures spatio-temporelles liées à ces modèles comme réponse à la requête. Toutefois, nous signalons que l'indexation et le codage ont été réalisés sur la troisième bande spectrale qui est la plus informative (cf. Tableau 6.6). Pour donner une meilleure interprétation visuelle à nos résultats, nous affichons les résultats sur les images de couleurs. Nous présentons ces résultats visuels de recherche par exemple dans les Figures 6.20, 6.21, 6.22 pour valider notre extraction d'information. Les résultats ne sont pas parfaits, mais en général les réponses données par le système sont en adéquation avec l'exemple présenté. De plus il est nécessaire de préciser, que les résultats de requête sont affichées sur 0.8% de la STIS totale. En conclusion, cette dernière expérience valide notre méthodologie pour l'extraction d'information spatio-temporelle des STIS. Nous notons que les résultats présentés sont largement meilleurs visuellement que ceux obtenus avec la méthodologie fondée sur le principe IB. En effet, dans ce dernier cas, l'information d'importance est connue à priori. Dans l'autre cas, c'est l'algorithme qui se charge de déterminer l'information d'intérêt.

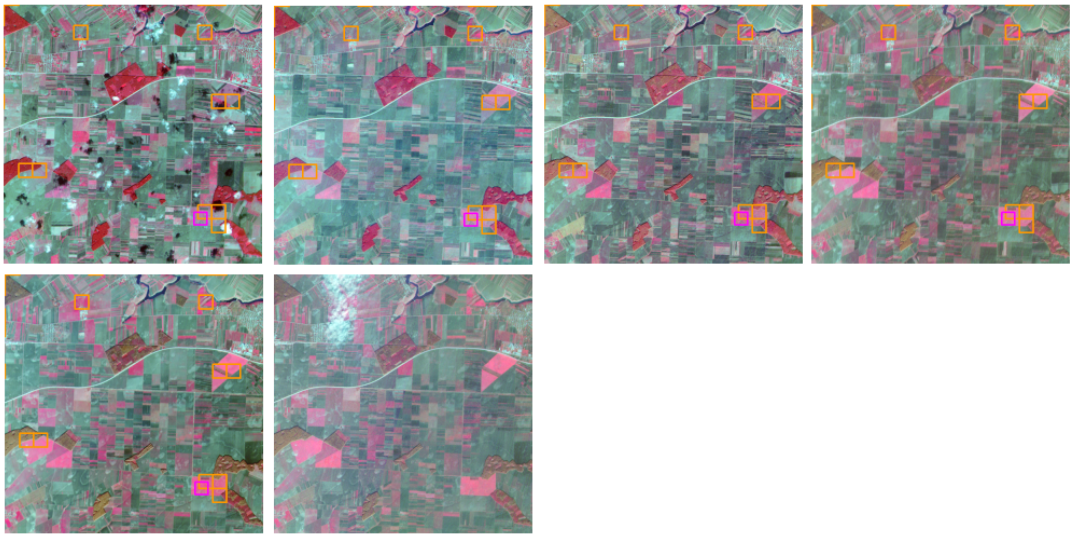


FIG. 6.21 – Dans cet exemple, nous avons voulu retrouver les champs qui évoluent d'un rose foncé tirant sur le vert, vers un rose pâle. Ce phénomène correspond à la maturation d'un certain type de culture. La sous série présentée est de taille $600 \times 600 \times 6$ prise au tout début de la STIS ADAM. Les résultats donnés présentent le même phénomène de maturation. Néanmoins, nous sommes forcé de constater que des événements du même type et non détectés apparaissent dans cette même séquence.

6.3 Résumé

Nous avons validé nos méthodologies sur des données synthétiques. En effet, ces données générées nous ont permis d'évaluer les résultats de clustering obtenus en les comparant à une classification de référence. De plus lors de ces évaluations, nous avons mis en évidence que la détermination du nombre optimal de clusters était souvent très proche du nombre de classes.
D'autre part, nous avons démontré l'intérêt de la méthode fondée sur le principe de MIB pour extraire de l'information tout en fusionnant différents types d'information.
Ensuite, nous avons validé la méthodologie fondée sur les complexités, en la comparant à des algorithmes d'extraction d'information de références. Nous avons effectué nos comparaisons à partir de mesures de qualité objectives couramment utilisées en indexation.
Enfin, nous avons évalué visuellement les résultats d'indexation en implémentant un système de fouille par exemple interagissant avec la STIS codée et indexée. Visuellement, les résultats sont satisfaisants.

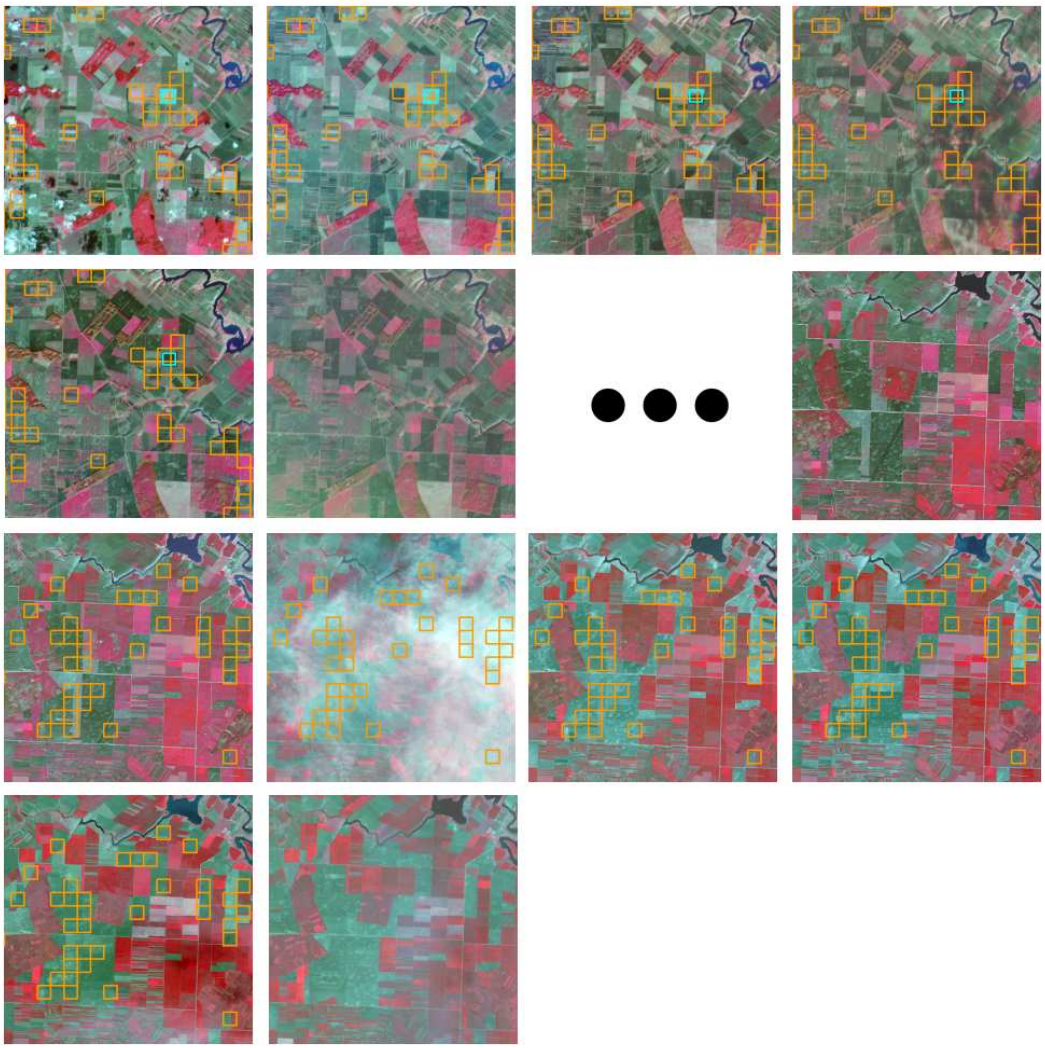


FIG. 6.22 – Cette série est divisée en deux parties. La première tranche temporelle est représentée sur les 6 premières images. Ensuite, quelques images de la STIS sont évincées, avant de continuer sur une seconde tranche temporelle composée de 7 images. Il y a une translation spatiale entre les deux tranches temporelles pour une meilleure visualisation des résultats. Par l'exemple représenté en cyan, nous avons souhaité retrouver les événements correspondant à la récolte des champs qui se traduit visuellement par une transition progressive du vert clair vers le vert foncé. Nous retrouvons ce phénomènes à deux endroits éloignés de la STIS. Ces résultats montrent l'intérêt de la fouille par le contenu pour localiser spatio-temporellement certains phénomènes.

Conclusion

Dans cette thèse, nous avons proposé de nouvelles méthodologies pour analyser et compresser conjointement les nombreux événements spatio-temporels qui apparaissent dans les STIS. Nous avons proposé un nouveau concept pour l'extraction d'information des STIS emprunté à la théorie du codage.

Synthèse des travaux

Après une analyse de l'état de l'art de l'extraction d'information et de système de recherche par le contenu, des liens forts entre l'extraction et la compression ont été mis en évidence. Cette constatation nous a amené à considérer un nouveau concept pour l'extraction d'information. Ce concept propose de considérer l'extraction d'information comme un problème de transmission d'information entre les données et un utilisateur. Dans cette vision du problème, nous voulons former un système d'extraction d'information le plus générique possible qui puisse être utilisé pour des applications RIBC, de recherche interactive ou de découvertes de connaissance. Cette approche nous a amené à penser que le codage et la compression peuvent fournir une représentation du contenu informationnel (le code) qui soit compacte et qui permette d'accéder facilement au contenu.
Par une description détaillée des différents cadres théoriques, tels que les statistiques bayésiennes, le codage entropique avec pertes et le codage de Kolmogorov, nous sommes arrivés à mettre en évidence les liens qui unissent ces points de vue à l'extraction d'information. Cette comparaison des cadres théoriques nous a permis de construire un cadre unifié pour l'extraction d'information. Nous avons déduit nos méthodologies pour l'extraction d'information au sein de ce cadre théorique.
Dans un premier temps, nous avons présenté des méthodologies issues de la théorie débit-distorsion pour extraire l'information sous forme compacte et parcimonieuse. D'une part, nous avons présenté le principe d'*Information Bottleneck* pour l'extraction d'information d'une variable aléatoire. Ce principe a l'avantage de qualifier l'information extraite par l'utilisation d'un variable aléatoire modélisant l'information d'importance. En rapport avec l'inférence bayésienne en deux niveaux, nous avons choisi de faire porter l'information pertinente par les modèles. Ce choix nous a permis d'intégrer un sélection de modèles floue dans la méthode d'extraction fondée sur le principe IB. Le principe IB permet d'obtenir la courbe débit-distorsion faisant le compromis entre la quantité d'information extraite et la quantité d'information d'importance. Pour choisir la quantité d'information extraite optimale et la plus universelle, nous avons introduit un critère heuristique fondé sur la courbure des fonctions débit-distorsion. Ce critère, expérimenté sur des techniques de clustering, permet d'identifier un couple (R, D) de la courbe débit-distorsion qui détermine un nombre optimal de clusters. Ainsi, la jonction de l'algorithme

IB et du critère de courbure permet de créer une méthodologie d'extraction automatique d'information d'importance. Pour appliquer cette méthodologie aux STIS, nous avons considéré que l'information d'importance était liée aux structures spatio-temporelles ce qui nous a amené à utiliser des champs aléatoires de Gibbs-Markov comme modèles. Ces champs aléatoires, nous apportent de l'information sur la complexité des dépendances spatio-temporelles et de ce fait sur l'évolution des textures. D'autre part, nous avons introduit le principe de *Multi-Information Bottleneck*. Ce principe est une extension du principe d'*Information Bottleneck* et il permet de prendre en compte plusieurs types d'information indépendantes. Dans le cas des STIS, ce principe nous est utile pour à la fois prendre en compte l'information de couleur et l'information de l'évolution des textures. Etant donné que l'information de couleur et de texture sont indépendants lors de la formation des données, le principe MIB est complètement adapté pour l'extraction de ces deux informations des STIS. Nous présentons une nouvelle fois comment le critère de courbure et l'algorithme MIB permettent d'extraire automatiquement une quantité optimale. Toutefois, nous avons noté que ce dernier principe permettait de privilégier certains types d'information lors de l'extraction. Enfin, nous avons décrit le principe de *Variational Information Bottleneck* qui est dual du principe IB. Nous avons calculé les équations consistantes qui permettraient de construire un algorithme. Nous présentons ce principe comme une nouvelle manière d'extraire en considérant cette fois-ci de l'information de désintérêt.

Dans un second temps, nous avons présenté une nouvelle méthode d'extraction fondée sur les complexités de Kolmogorov. En exploitant l'ambivalence du codage sans perte en deux parties (compression avec pertes et sans perte), nous avons pu construire une méthodologie pour construire un code incluant une représentation parcimonieuse de l'information d'intérêt. En fait, cette approche permet d'identifier l'information pertinente au sein de l'information totale contenue dans un ensemble d'objets informatifs. Tout d'abord, nous avons introduit la Distance Normalisée Suffisante de Compression (DNSC) pour comparer deux objets en exploitant l'information structurante des objets. Nous avons déduit cette distance de la DCN qui présente de très bons résultats pour le clustering de différents types d'objets, tels que le texte, les textures ou l'ADN. La DNSC est pertinente dans le cadre de l'extraction d'information, puisqu'elle intègre le principe LDM qui vise à extraire l'information d'intérêt. Ensuite, nous avons présenté un critère général de codage d'un ensemble d'objets en exploitant les redondances inter objets. Pour minimiser ce critère de longueur de code, nous avons créé une méthode d'optimisation sous-optimale fondée sur la DNSC et sur la décomposition de l'information en deux parties par un codeur universel. Cette méthode permet de construire un index des modèles. D'une part, cet index contient l'information pertinente extraite, d'autre part il est intégré dans le code en deux parties de l'ensemble d'objets. Enfin, dans le but d'appliquer notre méthodologie d'extraction aux STIS, nous avons créé un codeur sans perte pour les signaux tridimensionnels. Nous avons exploité l'expérience scientifique accumulée sur le codage d'image (Weinberger et al., 2000; Wu & Memon, 1997) pour construire ce codeur universel. Plus le codage est efficace, plus le modèle sous-jacent explique bien les données codées. Dans le cadre des STIS, ce codeur universel permet de produire un code des STIS qui inclut un index informationnel des structures spatio-temporelles.

Dans un troisième temps, nous avons validé nos méthodologies sur des données synthétiques et nous les avons expérimentées sur les STIS. Les données générées nous ont permis d'évaluer les résultats de clustering obtenus en les comparant à une classification de référence. De plus lors de ces évaluations, nous avons mis en évidence que la

détermination du nombre optimal de clusters était souvent très proche du nombre de classes. Ensuite, nous avons validé la méthodologie fondée sur les complexités, en la comparant à des algorithmes d'extraction d'information de références. Nous avons effectué nos comparaisons avec des mesures de qualité objectives couramment utilisées en indexation. Enfin, nous avons évalué visuellement les extractions d'information des STIS réalisées par les méthodologies IB, MIB et fondée sur les complexités.
Le concept et les méthodologies développées dans cette thèse constituent une nouvelle approche prometteuse pour l'extraction d'information en général et dans les STIS en particulier. Elles ont l'avantage de considérer l'information en tant que telle.

Perspectives

Plusieurs axes intéressants de recherche et de développement se dégagent de ces travaux de thèse.
D'abord nous résumons les points de développement à apporter à cette thèse. Dans le cadre des méthodologies IB, il faudrait prendre en compte l'information géométrique pour avoir les trois types d'information : couleur, texture et géométrie. Par exemple prendre en compte cette information permettrait d'une part des résultats visuels plus probant, d'autre part permettrait d'améliorer l'extraction. Il serait par exemple plus judicieux de partitionner la STIS en champs aléatoires dont les formes épousent les objets visibles. Dans le cadre de la méthodologie fondée sur le codeur, il faudrait incorporer l'information de couleur en utilisant des transformées de décorrelation spectrales lors du codage par exemple. D'autre part, l'information géométrique pourrait être incorporée spatialement et temporellement par des techniques de segmentation fondée sur les longueurs de code. En effet dans les critères LDM mis en place, le nombre d'échantillons codés est un argument invariable. En considérant des formes plus complexes des champs aléatoires, ce nombre d'échantillons rentrerait en compte pour minimiser les critères LDM. Comme première étape, la segmentation temporelle serait vite adaptable au codeur mis en place et donnerait une meilleure extraction au niveau temporelle. Enfin, comme les série temporelles constituent un volume de données en constante croissance, il faudrait pouvoir coder efficacement les nouvelles images. Dans le cadre du chapitre 5, il serait facile d'imaginer un procédure pour intégrer au code les nouveaux événements spatio-temporels. Par exemple, quand un nouvel objet doit être codé, il suffit de comparer les longueurs de code soit en l'intégrant dans un groupe dont le modèle est connu, soit en constituant un nouveau groupe dont l'objet est le seul élément. Cette démarche permettrait de continuer à absorber dans le code de nouvelles images d'une STIS, tout en faisant évoluer l'index en y ajoutant les nouveautés.
L'évaluation d'un système d'extraction n'est pas aisée. La procédure de validation des méthodologies mise en place est trop simple pour valider les résultats sur les STIS. Il faudrait s'appuyer sur une procédure d'évaluation intensive qui doit être effectuée avec de grandes quantités de données de nature variées avec des caractéristiques diverses. Un moyen d'évaluer l'extraction d'information, serait de l'exploiter dans différentes applications intégrant le paradigme de la fouille, telles que la recherche interactive fondée sur les MVS, l'apprentissage ou la détection de nouveautés.
Ensuite, d'un point de vue théorique ce travail de recherche a ouvert de nouvelles perspectives pour l'extraction d'information non-supervisée. Premièrement, serait-il pertinent d'utiliser le principe de *Variational Information Bottleneck* pour extraire l'informa-

tion en connaissant l'information non importante ? D'ailleurs avant d'utiliser ce nouveau principe, il faudrait étudier la convergence des algorithmes qui en découlent. Deuxièmement, concernant la méthodologie fondée sur les complexités, il serait intéressant de voir si elle peut s'appliquer à des systèmes de codage avec pertes. Par exemple, le codage par ondelettes utilise des modèles statistiques pour déterminer le meilleur compromis débit-distorsion à un débit donné. Troisièmement, pour constituer un système de recherche efficace, il est peut-être judicieux d'intégrer la complexité des recherches dans le design du système. Il est possible qu'il y est un compromis entre la complexité de recherche et la taille de stockage des données (cf. §2.4.1). Finalement, le rapprochement entre le paradigme de la fouille et la transmission d'information, pourrait permettre d'exploiter au mieux toute la connaissance qui a été accumulée en compression et codage de canal. Même plus, en s'approchant des techniques de codage conjoint source-canal, il serait possible d'obtenir de nouvelles méthodes au sein du paradigme de la fouille. Par exemple, si nous souhaitons comparer deux objets avec un distance connue, un problème est de savoir quelles sont les quantités d'informations minimales à transmettre des deux objets pour reconstruire la distance avec un certain degré de confiance (Doshi et al., 2007).
Finalement, d'un point de vue applicatif, les méthodologies employées pourraient permettre d'inventer de nouveaux capteurs intelligents qui se concentreraient sur des zones d'intérêt ou mêmes des événements d'intérêts en ne transmettant que le code de ces zones.

Glossaire

ADAM : Assimilation de Données par Agro-Modélisation
CNES : Centre National d'Etude Spatiale
DCN : Distance de Compression Normalisée
DNSC : Distance Normalisée Suffisante de Compression
GMES : *Global Monitoring for Environment and Security*
GRACE : *Gravity Recovery and Climate Experiment*
IB : *Information Bottleneck*
JPEG : *Joint Photographic Experts Group*
KIM : *Knowledge driven Information Mining*
KL : Kullback-Leibler
LDM : Longueur de Description Minimale
MAP : Maximum A Posteriori
MCL : Métrique de Chen-Li
MDC : Mesure de Dissimilitude basée sur la Compression
MDIC : Modulation Différentielle d'Impulsions Codées
MEQM : Minimum de l'Erreur Quadratique Moyenne
MIB : *Multi-Information Bottleneck*
MPEG : *Moving Picture Experts Group*
MV : Maximum de Vraisemblance
MVS : Machines à Vecteurs Supports
NASA : *National Aeronautics and Space Administration*
PV : Pseudo-Vraisemblance
QBIC : *Query By Image Content*
RIBC : Recherche d'Images Basée sur le Contenu
RSO : Radar à Synthèse d'Ouverture
SFI : Systèmes de Fonctions Itérées
SPOT : Satellite Pour l'Observation de la Terre
STIS : Série Temporelles d'Images Satellitaires
TCD : Transformée en Cosinus Discret
TSV : Teinte Saturation Valeur
VIB : *Variational Information Bottleneck*

Bibliographie

Akaike, H. (1954). An Approximation to the Density Functions, *Annales of Institute on Statistics and Mathematics* **6** : 127–132.

Aurdal, L., Huseby, R. B., Eikvil, L., Solberg, R., Vikhamar, D. & Solberg, A. (2005). Use of Hidden Markov Models and Phenology for Multitemporal Satellite Image Classification :Applications to Mountain Vegetation Classification, *International Workshop on the Analysis of Multi-Temporal Remote Sensing Images*, Biloxi, USA, pp. 220–224.

Bader, D., Jaja, J. & Chellappa, R. (1995). Scalable Data Parallel Algorithms for Texture Synthesis Using Gibbs Random Fields, *IEEE Transactions on Image Processing* **4**(10) : 1456–1461.

Banerjee, A., Dhillon, I., Ghosh, J. & Merugu, S. (2004a). An information theoretic analysis of maximum likelihood mixture estimation for exponential families, *ACM Twenty-first international conference on Machine learning*, Vol. 8, ACM Press, Alberta, Canada.

Banerjee, A., Merugu, S., Dhillon, I. & Ghosh, J. (2004b). Clustering with Bregman Divergences, *SIAM International Conference on Data Mining*.

Barron, J., Fleet, D. & Beauchemin, S. (1994). Performance of optical flow techniques, *International Journal on Computer Vision* **12**(1) : 43–77.

Bazi, Y., Bruzzone, L. & Melgani, F. (2005). An Unsupervised Approach Based on the Generalized Gaussian Model to Automatic Change Detection in Multitemporal SAR Images, *IEEE Transactions on Geoscience and Remote Sensing* **43**(4) : 874–887.

Beckmann, N., Kriegel, H.-P., Schneider, R. & Seeger, B. (1990). The R*-tree : An efficient and robust access method for points and rectangles, *Proc. ACM SIGMOD Int. Conf. Management Data*, Vol. 19, pp. 322–331.

Bennett, C., Gacs, P., Li, M., Vitanyi, P. & Zurek, W. (1998). Information Distance, *IEEE Transactions on Information Theory* **44**(7) : 1407–1423.

Bezdek, J. (1981). *Recognition with Fuzzy Objective Function Algorithms*, New York Plenum Press.

Blahut, R. (1972). Computation of Channel Capacity and Rate-Distortion Functions, *IEEE Transactions on Information Theory* **IT-18**(4) : 460–473.

Boser, B., Guyon, I. & Vapnik, V. (1992). A Training Algorithm for Optimal Margin Classifiers, *5th Annual ACM Workshop on Computational Learning Theory*, New York, USA, pp. 144–152.

Bovolo, F. & Bruzzone, L. (2005). A detail-preserving scale-driven approach to change detection in mutlitemporal sar images, *IEEE Transactions on Geoscience and Remote Sensing* **43**(12) : 2963–2972.

Bunke, H. & Allerman, G. (1983). Inexact Graph Matching for Structural Pattern Recognition, *Pattern Recognition Letters* **1**(4) : 245–253.

Campedel, M., Luo, B., Maitre, H., Moulines, E., Roux, M. & Kyrgyzov, I. (2004). Indexation des images satellitaires : Détection et évaluation des caractéristiques de classification, *Technical Report ENST2004D008*, École Nationale Supérieure des Télécommunications, Paris, France.

Carson, C., Belongie, S., Greenspan, H. & Malik, J. (2002). Bloworld :Image Segmentation Using Expecation-Maximization and Its Application to Image Query, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(8) : 1026–1038.

Chang, S.-F., Chen, W., Meng, H., Sundaram, H. & Zhong, D. (1998). A Fully Automated Content-Based Video Search Engine Supporting Spatiotemporal Queries, *IEEE Transactions on Circuits and Systems for Video Technology* **8** : 102–615.

Chang, T. & C.-C.J., K. (1993). Texture Analysis and Classification with Tree-Structured Wavelet Transform, *IEEE Transactions on Image Processing* **2**(4) : 429–441.

Charikar, M., Chekuri, C., Feder, T. & Motwani, R. (1997). Incremental Clustering and Dynamic Information Retrieval, *29th Annual ACM Symposium on Theory of Computing*, El Paso, USA, pp. 626–635.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W. & Freeman, D. (1988). Autoclass : A bayesian classification system, *Fifth International Conference on Machine Learning*, pp. 54–64.

Chellappa, R. & Kashyap, R. (1983). Estimation and Choice of Neighbors in Spatial-Interaction Models of Images, *IEEE Transactions on Information Theory* **IT-29**(1) : 60–73.

Chellappa, R. & Kashyap, R. (1985). Texture Synthesis Using 2-D Noncausal Autoregressive Models, *IEEE Transactions on Acoustics, Speech and Signal Processing* **ASSP-33**(1) : 194–204.

Chen, X., Kwong, S. & Li, M. (1999). A Compression Algorithm for DNA Sequences and Its Applications in Genome Comparison, *10th Workshop on Genome Informatics*, Tokyo, Japan, pp. 51–61.

Choi, H. & Baraniuk, R. (2001). Multiscale Image Segmentation Using Wavelet-Domain Hidden Markov Models, *IEEE Transactions on Image Processing* **10**(9) : 1309–1321.

Clarke, B. & Barron, A. (1990). Information-Theoric Asymptotics of Bayes Methods, *IEEE Transactions on Information Theory* **36**(3) : 453–471.

Comaniciu, D. & Meer, P. (2002). Mean Shift :A Robust Approach Toward Feature Space Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5) : 603–619.

Cortes, C. & Vapnik, V. (1995). Support Vector Networks, *Machine Learning* **20** : 1–25.

Costache, M., Maître, H. & Datcu, M. (2006). Categorization based Relevance Feedback Search Engine for Earth Observation Images Repositories, *IEEE International Geoscience and Remote Sensing Symposium*, Denver, USA, pp. 13–16.

Cover, T. & Hart, P. (1967). Nearest Neighbor Pattern Classification, *IEEE Transcations on Information Theory* **IT-13** : 21–27.

Cover, T. & Thomas, J. (1991). *Elements of Information Theory*, Wiley-Interscience.

Cross, G. & Jain, A. (1983a). Markov random field texture models, *IEEE Transactions on Pattern Recognition and Machine Intelligence* **5**(1) : 25–39.

Cross, G. & Jain, A. (1983b). Markov Random Field Texture Models, *IEEE Transactions Pattern Analysis and Machine Intelligence* **5**(1) : 25–39.

Datcu, M., Daschiel, H. & al. (2003). Information Mining in Remote Sensing Image Archives : System Description, *IEEE Transaction on Geoscience and Remote Sensing* **41**(12) : 2923–2936.

Datcu, M., Seidel, K. & Walessa, M. (1998). Spatial Information Retrieval from Remote-Sensing Images. i. Information Theoritical Perspectives, *IEEE Transactions on Geoscience and Remote Sensing* **36**(5) : 1431–1445.

Daubechies, J. (1990). The Wavelet Transformation, Time-Frequency Localization and Signal Processing, *IEEE Transactions on Information Theory* **36**(5) : 961–1005.

Dawid, A. (1987). Estimation and Inference by Compact Coding, *Journal Royal Statistics Society* **B 49** : 253–254.

Dempster, A., Laird, N. & Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society* **B 39**(1) : 1–38.

Devroye, L. (1978). A Universal k-Nearest Neighbor Procedure in Discrimination, *IEEE Computer Society Conference on Pattern Recognition and Image Processing*, Long Beach, USA, pp. 142–147.

Do, M. & Vetterli, M. (2002). Rotation Invariant Texture Characterization and Retrieval Using Steerable Wavelet-Domain Hidden Markov Models, *IEEE Transactions on Multimedia* **4**(4) : 517–527.

Do, M. & Vetterli, M. (2003). *Beyond Wavelets*, Academic Press, chapter Contourlets.

Doshi, V., Shah, D., Médard, M. & Jaggi, S. (2007). Distributed Functional Compression through Graph Coloring, *IEEE Data Compression Conference*, Snowbird, USA, pp. 93–102.

Duda, R., Hart, P. & Stork, D. (2001). *Pattern Classification*, Wiley.

Farach, M. & Thorup, M. (1998). String Matching in Lempel-Ziv Compressed Strings, *Algorithmica* **4**(20) : 388–404.

Fisher, R. (1936). The Use of Multiple Measurements in Taxonomic Problems, *Ann. Eugenics* **7** : 179–188. 110.

Flickner, M., Sawhney, H., Niblack, W. & al. (1995). Query by Image and Video Content : The QBIC System, *IEEE Computer* **28** : 23–32.

Friedman, N., Mozenzon, O., Slonim, N. & Tishby, N. (2001). Multivariate Information Bottleneck, *Proceedings of Uncertainty in Artificial Intelligence*, San Fransisco, USA, pp. 152–161.

Fuglede, B. & Topsoe, F. (2004). Jensen-Shannon Divergence and Hilbert Space Embedding, *International Symposium on Information Theory*, pp. 31–41.

Fukunaga, K. & Flick, T. (1984). An Optimal Global Nearest Neighbor Metric, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6** : 314–318.

Geman, S. & Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(6) : 721–741.

Giros, A. (2006). Comparison of Partitions of Two Images for Satellite Images Time Series Segmentation, *IEEE International Geoscience And Remote Sensing Symposium*, Denver, USA, pp. 2592–2595.

Globerson, A. & Tishby, N. (2003). Sufficient Dimension Reduction, *Journal of Machine Learning Reserach* **3** : 1307–1331.

Goldberger, J., Greenspan, H. & Gordon, S. (2002). Unsupervised Image Clustering Using the Information Bottleneck Method, *24th DAGM Symposium, Pattern Recognition*, Zurich, Switzerland, pp. 158–165.

Gray, R. & Neuhoff, D. (1998). Quantization, *IEEE Transactions on Information Theory* **44**(6) : 2325–2383.

Grossi, R. & Vitter, J. (2005). Compressed Suffix Arrays and Suffix Trees with Applications to Text Indexing and String Matching, *SIAM Journal on Computing*.

Grünwald, P. & Vitanyi, P. (2004). Shannon Information and Kolmogorov Complexity, Amsterdam, the Netherlands.

Gueguen, L. & Datcu, M. (2006). Spatio-Temporal Structures Characterization Based on Multi-Information Bottleneck, *ESA-EUSC 2006 : Image Information Mining for Security and Intelligence*, Madrid.

Gueguen, L. & Datcu, M. (2007a). Image Time-Series Data Mining Based on the Information-Bottleneck Principle, *IEEE Transactions on Geoscience and Remote Sensing* **45**(4) : 827–838.

Gueguen, L. & Datcu, M. (2007b). The Model Based Similarity Metric, *IEEE Data Compression Conference*, Snowbird, USA, p. 382.

Gueguen, L., Le Men, C. & Datcu, M. (2006). Analysis of Satellite Image Time Series Based on Information Bottleneck, *26th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, Paris, France, pp. 367–374.

Gueguen, L., Trocan, M., Pesquet-Popescu, B., Giros, A. & Datcu, M. (2005). Comparison of Multispectral Satellite Sequence Compression Approaches, *International Symposium on Signals, Circuits and Systems*, Vol. 1, Iasi, Romania, pp. 87–90.

Hanjalic, A., Lagendijk, R. & Biemond, J. (1999). Automated High-Level Movie Segmentation for Advanced Video-Retieval Systems, *IEEE Transactions on Circuits and Systems for Video Technology* **9**(4) : 580–588.

Hansen, P. & O'Leary, D. (1993). The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems, *SIAM Journal on Scientific Computing* **14**(6) : 1487–1503.

Haralick, R., Shanmugan, K. & Dinstein, I. (1973). Textural Features for Image Classification, *IEEE Transactions on Systems, Man and Cybernetics* **6**(3) : 610–621.

Heas, P. (2005). *Apprentissage Bayésien de Structures Spatio-Temporelles :"application à la fouille visuelle de séries temporelles d'images de satellites*, PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace.

Heas, P. & Datcu, M. (2005). Modelling Trajectory of Dynamic Cluster in Image-Time-Series for Spatio-Temporal Reasoning, *IEEE Transactions on Geoscience and Remote Sensing* **43**(7) : 1635–1647.

Huang, J., Kumar, S., Mitra, M., Zhu, W. & Zabih, R. (1997). Image Indexing using Color Correlograms, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 762–768.

Idris, F. & Panchanathan, S. (1995). Image Indexing Using Wavelet Vector Quantization, *Proceedings SPIE on Digital Image Storage Archiving Systems*, Vol. 2606, pp. 269–275.

Jacquin, A. (1993). Fractal Image Coding :A Review, *Proceedings of IEEE*, Vol. 81, pp. 1451–1465.

Jain, A. & Farrokhnia, F. (1991). Unsupervised Texture Segmentation using Gabor Filters, *Pattern Recognition* **12**(24) : 1167–1186.

Jaynes, E. (1968). Prior Probabilities, *IEEE Transactions On Systems Sceience and Cybernetics* **4**(3) : 227–241.

Jeffreys, H. (1948). *Theory of Probability*, Oxford : Clarendon Press, chapter 1,3, pp. 158–167.

Jiang, J., Liu, M. & Hou, C. (2003). Texture-Based Image Indexing in the Process of Lossless Data Compression, *Proceedings on Vision, Image and Signal Processing*, Vol. 150, pp. 198–204. historgram of states dans un flux compresse par JPEG-LS.

Jing, X., Liu, L., Zhang, C., Li, X., Li, Y. & Jia, J. (2005). The Extraction of Beijing Main Crops Planting Area Based on Time Series MODIS NDVI Reconstruction, *IEEE Geoscience and Remote Sensing Symposium*, Vol. 4, Seoul, Korea, pp. 3020–3023.

Johnson, S. (1967). Hierarchical Clustering Schemes, *Psychometrika* **32**(3) : 241–254.

Kaufman, L. & Rousseeuw, P. (1990). *Finding Groups in Data*, John Wiley and Sons, New York.

Kawamura, M., Tsujiko, Y., Tsujino, K. & Sakai, T. (2004). Time-Series Fire-Induced Forest Hazard Mapping Using Landsat and IKONOS Imageries, *IEEE Geoscience and Remote Sensing Symposium*, Vol. 4, Anchorage, USA, pp. 2256–2259.

Keim, D. & Kriegel, H.-P. (1996). Visualization Techniques for Mining Large Databases : A Comparison, *IEEE Transactions on Knowledge and Data Engineering* **8**(6) : 923–938. paradigme de la fouille d'information.

Keogh, E., Lonardi, S. & Rtanamahatana, C. A. (2004). Toward Parameter-Free Data Mining, *10th ACM SIGKDD*, Seattle, USA, pp. 206–215.

Kobla, V., DeMenthon, D. & Doermann, D. (1999). Detection of Slow-Motion Replay Sequences for Indentifying Sports Videos, *IEEE Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, pp. 135–140.

Kobla, V., Doermann, D., Lin, K.-I. & Faloutsos, C. (1997). Compressed Domain Video Indexing Techniques Using DCT and Motion Vector Information in MPEG Video, *SPIE conference on Storage and Retrieval for Image and Video Databases*, Vol. 3022, San Jose, USA, pp. 200–211.

Kolmogorov, A. (1965). Three Approaches to the Quantitative Definition of Information, *Problems of Informations Transmission* **1**(1) : 1–7.

Koprinska, I. & Carrato, S. (2001). Temporal Video Segmentation :A Survey, *Signal Processing and Image Communication* **16** : 451–460.

Kullback, S. & Leibler, R. (1951). On Information and Sufficiency, *The Annals of Mathematical Statistics* **22**(1) : 79–86.

Laine, A. & Fan, J. (1993). Texture Classification by Wavelet Packet Signatures, *IEEE Transactions on Pattern Analyse and Machine Intelligence* **15**(11) : 1186–1191.

Lazebnik, S., Schmid, C. & Ponce, J. (2003). Affine-Invariant Local Descriptors and Neighborhood Statistics for Texture Recognition, *IEEE International Conference on Computer Vision*, Vol. 1, Nice, France, pp. 649–655.

Lee, T., Girolami, M., Bell, A. & Sejnowski, T. (2000). A Unifying Information-Theoretic Framework for Independant Component Analysis, *Computers and Mathematics with Applications* **31**(11) : 1–21.

Li, M. & Vitanyi, P. (1997). *An Introduction to Kolmogorov Complexity and Its Applications*, 2nd edition edn, Springer Verlag.

Li, M., Chen, X., Li, X., Ma, B. & Vitanyi, P. (2004). The Similarity Metric, *IEEE Transactions on Information Theory* **50**(12) : 3250–3264.

Li, X. (2003). On Exploiting Geometric Constraint of Image Wavelet Coefficient, *IEEE Transactions on Image Processing* **10**(11) : 1378–1387.

Li, Y. & Bretshneider, T. (2007). Semantic-Sensitive Satellite Image Retrieval, *IEEE Transactions on Geoscience and Remote Sensing* **45**(4) : 853–860.

Liu, F. & Picard, W. (1996). Periodicity, Directionality, and Randomness : Wold Features for Image Modeling and Retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(7) : 722–733.

Lucchese, L. & Mitra, S. (2001). Color Image Segmentation :A State-of-the-art Survey, *Proceedings of the Indian Ntional Science Academy*, pp. 207–221.

Luo, B., Aujol, J., Gousseau, Y., Ladjal, S. & Maître, H. (2006). Characteristic Scale in Satellite Images, *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, Toulouse, France, pp. 809–812.

MacKay, D. (1991). *Maximum Entropy and Bayesian Methods in Inverse Problems*, Kluwer Academic Publisher, chapter Bayesian Interpolation, pp. 39–66.

MacKay, D. (2003). *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press.

MacQueen, J. (1965). Some Methods for Classification and Analysis of Multivariate Observations, *5-th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, Berkeley, USA, pp. 281–297.

Mandal, M., Idris, F. & Panchanathan, S. (1999). A Critical Evaluation of Image and Video Indexing Techniques in the Compressed Domain, *Image and Vision Computing* **17**(7) : 513–529.

Manjunath, B. & Ma, W. (1996). Texture Features for Browsing and Retrieval of Image Data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(8) : 837–842.

Matas, J., Marik, R. & Kittler, J. (1995). On Representation and Matching of Multi-Coloured Objects, *IEEE International Conference on Computer Vision*, Boston, USA, pp. 726–732.

McCulloch, W. & Pitts, W. (1943). A Logical Calculus of the Ideas Imminent in Neural Activity, *Bulletin on Mathematics in Biophysics* **5** : 115–133.

McLean, G. (1993). Vector Quantization for Texture Classification, *IEEE Transactions on Systems, Man and Cybernetics* **23**(3) : 637–649.

Meila, M. (2003). Comparing Clustering by the Variation of Information, *Poceedings of the 16th Annual Conference of Computational Learning Theory*, Springer.

Minka, T. & Picard, R. (1997). Interactive learning using a society of models, *Pattern Recognition* **4**(30) : 565–581.

Mitra, P., Murthy, C. & Pal, S. (2002). Unsupervised Feature Selection Using Feature Similarity, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(3) : 301–312.

Mojsilovic, A., Hu, H. & Soljanin, E. (2002). Extraction of Perceptually Important Colors and Similarity Measurement for Image Matching, Retrieval and Analysis, *Transactions on Image Processing* **11**(11) : 1238–1248.

Netravali, A. & Limb, J. (1980). Picture Coding : A Review, *Proceedings of IEEE* **68** : 366–406.

O Ruanaidh, J. & Fitzgerald, W. (1996). *Numerical Bayesian Methods Applied to Signal Processing*, Springer, chapter 2.

Pajarola, R. & Widmayer, P. (1996). Pattern Matching in Compressed Raster Images, *Proceedings of the Third South American Workshop on String Processing*, pp. 228–242.

Pajarola, R. & Widmayer, P. (2000). An Image Compression Method for Spatial Data Search, *IEEE Transactions on Image Processing* **9**(3) : 357–365.

Pass, G., Zabih, R. & Miller, J. (1996). Comparing Images Using Color Coherence Vectors, *ACM Multimedia*, pp. 65–73.

Pentland, A., Picard, R. & Sclaroff, S. (1994). Photobook : Tools for Content-Based Manipulation of Image Databases, *SPIE, Storage and Retrieval for Image and Video Databases*, San Jose, USA, pp. 34–47.

Pi, M., Mandal, M. K. & Basu, A. (2005). Image Retrieval Based on Histogram of Fractal Parameters, *IEEE Transactions on Multimedia* **7**(4) : 597–605.

Podilchuk, C. & Zhang, X. (1996). Face Recognition Using DCT-Based Feature Vectors, *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 4, Atlanta, USA, pp. 2144–2147.

Popat, A. (1997). *Conjoint Probalistic Subband Modeling*, PhD thesis, Massachusetts Institute of Technology.

Puzicha, J., Hofmann, T. & Buhmann, J. (1997). Non-Parametric Similarity Measures for Unsupervised Texture Segmentation and Image Retrieval, *Conference on Computer Vision and Pattern Recognition*, Puerto Rico, p. 267.

Rahman, A. & Murshed, M. (2004). Real-time temporal texture characterization using block-based motion co-occurence statistics, *IEEE International Conference on Image Processing*, Vol. 3, Singapore, pp. 1593–1596.

Randen, T. & Husoy, J. (1995). Multichannel Filtering for Image Segmentation, *SPIE Optical Engineering* **8**(33) : 2617–2625.

Randen, T. & Husoy, J. (1999). Filtering for Texture Classification : A Comparative Study, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(4) : 291–.

Rissanen, J. (1983). A Universal Data Compression System, *IEEE Transactions on Information Theory* **IT-29**(5) : 656–664.

Rissanen, J. (1986). Stochastic Complexity and Modeling, *Annals of Statistics* **14**(3) : 1080–1100.

Rissanen, J. (n.d.). Lectures on Statistical Modeling Theory, Helsinki Institute for Information Technology.

Rochery, M., Jermyn, I. & Zerubia, J. (2003). High Order Active Contours and their Apllications to the Detection of Line Networks in Satellite Imagery, *IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision In Conjunction*.

Rose, K. (1998). Deterministic Annealing for Clustering, Compression, Classification, Regression and Related Optimization Problems, *Proceedings of IEEE* **86**(11) : 2210–2239.

Rubner, Y., Tomasi, C. & Guibas, L. (1998). A Metric for Distributions with Applications to Image Databases, *IEEE International Conference on Computer Vision*, Bombay, India, pp. 59–66.

Rucklidge, W. (1996). *Efficient Visual Recognition Using the Hausdorff Distance*, Springer-Verlag New York.

Schouten, B. & Zeeuw, P. (2000). Image Databases, Scale and Fractal Transforms, *International Conference on Image Processing*, Vol. 2, Vancouver, Canada, pp. 534–537.

Schroder, M., Rehrauer, H., Seidel, K. & Datcu, M. (1998). Spatial Information Retrieval from Remote-Sensing Images. ii. Gibbs-Markov Random Fileds, *IEEE Transactions on Geoscience and Remote Sensing* **36**(5) : 1446–1455.

Schröder, M. & Dimai, A. (1998). Texture Information in Remote Sensing Images : A Case Study, *Workshop on Texture Analysis*, Freiburg, Germany.

Sculley, D. & Brodley, C. (2006). Compression and Machine Learning : A New Perspective on Feature Space Vectors, *IEEE Data Compression Conference*, pp. 332–341.

Seidel, K., Mastropietro, R. & Datcu, M. (1997). New Architectures for Remote Sensing Images Archives, *IGARSS '97*, Vol. 1, pp. 616–618. fondations de la fouille d'information.

Shannon, C. (1948). A Mathematical Theory of Communication, *The Bell System Technical Journal* **27** : 379–423,623–656.

Shi, J. & Malik, J. (2000). Normalized Cuts and Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8) : 888–905.

Shyu, C.-R., Klaric, M., Scott, G., Barb, A., Davis, C. & Palaniappan, K. (2007). GeoIRIS :Geospatial Information Retrieval and Indexing System-Content Mining, Semantics Modeling, and Complex Queries, *IEEE Transactions on Geoscience and Remote Sensing* **45**(4) : 839–852.

Skarbek, W. & Koschan, A. (1994). Color image segmentation : A survey, *Technical report*, Berlin University.

Smith, J. & Chang, S.-F. (1997). VisualSEEk : a fully automated content-based image query system, *in* I. M. Conference (ed.), *Proceedings of fourth ACM international conference on Multimedia*, ACM Press, pp. 87–98.

Spitzer, F. (1971). Markov Random Field and Gibbs Ensembles, *The American Mathematical Monthly* **78**(2) : 142–154.

Stoian, R. (1988). *Compresie de date, algoritmi de predictie*, Edictura Stiintificalsi Enciclopedica, Bucuresti.

Stricker, M. & Orengo, M. (1995). Similarity of Color Images, *SPIE Storage and Retrieval for Image and Video Databases*, pp. 381–392.

Sugar, C. & James, G. (1998). Finding the Number of Clusters in a DataSet : An Information Theoretic Approach, *Journal of the American Statistical Association* pp. 750–763.

Swain, M. & Ballard, D. (1991). Color Indexing, *International Journal of Computer Vision* **1**(7) : 11–32.

Swanson, M., Hosur, S. & Tewfik, A. (1996). Coding for Content-Based Retrieval, *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 4, Atlanta, USA, pp. 1959–1962.

Tabesh, A., Bilgin, A., Krishnan, K. & Marcellin, M. W. (2005). JPEG2000 and Motion JPEG2000 Content Analysis Using Codestream Lenght Information, *Data Compression Conference*, IEEE Computer Society, Snowbird, USA, pp. 329–337.

Taneja, I. (2001). *Generalized Information Measures and Their Applications*, online : http ://www.mtm.ufsc.br/ taneja/book/book.html.

Tishby, N., Pereira, F. & Bialek, W. (1999). The Infomation Bottleneck Method, *Proc 37th Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377.

Vapnik, V. (1982). *Estimation of Dependencies Based on Empirical Data*, Springer-Verlag.

Veltkamp, R. & Hagedoorn, M. (1999). State-of-the-art in shape matching, *Technical Report UU-CS-1999-27*, Utrecht University, Netherlands.

Vereshchagin, N. & Vitanyi, P. (2004). Kolmogorov's Structure Functions and Model Selection, *IEEE Transactions on Information Theory* **50**(12) : 3265–3290.

Wagner, R. & Fisher, M. (1974). The String-to-String Correction Problem, *Journal of the ACM* **20**(1) : 168–173.

Wang, H., Divakaran, A., Vetro, A., Chang, S.-F. & Sun, H. (2003). Survey of Compressed-Domain Features Used in Audio-Visual Indexing and Analysis, *Journal of Visual Communication and Image Representation* **14** : 150–183.

Wang, J., Li, J. & Wiederhold, G. (2001). SIMPLIcity :Semantic-Sensitive Integrated Matching for Picture LIbraries, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(9) : 947–963.

Wang, Y. & Loe, K. a. (2005). Spatiotemporal Video Segmentation Based on Graphical Models, *IEEE Transactions on Image Processing* **14**(7) : 937–947.

Weinberger, M., J.J., R. & R.B., A. (1996a). Applications of Universal Context Modeling to Lossless Compression of Gray-Scale Images, *IEEE Transactions on Image Processing* **5**(4) : 575–586.

Weinberger, M., Seroussi, G. & Sapiro, G. (1996b). LOCO-I : A Low Complexity, Content-Based, Lossless Image Compression Algorithm, *IEEE Data Compression Conference*, Snowbird, USA, pp. 140–149.

Weinberger, M., Seroussi, G. & Sapiro, G. (2000). The LOCO-I Lossless Image Compression Algorithm : Principles and Standardization into JPEG-LS, *IEEE Transactions on Image Processing* **9**(8) : 1309–1324.

Welch, T. (1984). A technique for high performance data compression, *j-Computer* **17**(6) : 8–20.

Winkler, G. (1995). *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, Springer Verlag.

Wornell, G. (1993). Wavelet-Based Representations for the 1/f Family of Fractal Processes, *Proceedings of the IEEE* **81**(10) : 1428–1450.

Wu, X. & Memon, N. (1997). Context-Based, Adaptive, Lossless Image Coding, *IEEE Transactions on Communications* **45**(4) : 437–444.

Yueting, Z., Yong, R., Huang, T. & Mehrotra, S. (1998). Adaptive Key Frame Extrcation Using Unsupervised Clustering, *IEEE International Conference on Image Processing*, Vol. 1, Chicago, USA, pp. 866–870.

Zalesny, A., Ferrari, V., Caenen, G. & Van Gool, L. (2002). Parallel Composite Texture Synthesis, *Texture 2002 Workshop in conjunction with ECCV 2002*, Copenhagen, Denmark, pp. 151–155.

Zhang, A., Cheng, B. & Acharya, R. (1995). Approach to query-by-texture in image database systems, *SPIE on Digital Image Storage and Archiving Systems*, Vol. 2606, pp. 338–349.

Zhu, L., Rao, A. & Zhang, A. (2000). Keyblock :An Approach for Content-Based Geographic Image Retrieval, *First Interantional Conference on Geographic Information Science*, Savannah, USA.

Zvonkin, A. & Levin, L. (1970). The complexity of finite objetcs and the development of the concepts of information and randomness by means of the theory of algorithms, *Russian Mathematical Surveys* **25**(6) : 83–124.