HAL
open science

Description multiple de l'information par transformation Mojette

Benoît Parrein

To cite this version:

Benoît Parrein. Description multiple de l'information par transformation Mojette. Interface homme-machine [cs.HC]. Université de Nantes, 2001. Français. NNT : . tel-00300613

HAL Id: tel-00300613

https://theses.hal.science/tel-00300613v1

Submitted on 18 Jul 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.

UNIVERSITÉ DE NANTES

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES
DE L'INFORMATION ET DES MATERIAUX

Année 2001 N° B.U. :

Thèse de Doctorat de l'Université de Nantes

Spécialité Électronique

Option : Traitement d'images

Présentée et soutenue publiquement par

Benoît PARREIN

le 22 Novembre 2001.

à l'École polytechnique de l'université de Nantes

DESCRIPTION MULTIPLE DE L'INFORMATION PAR TRANSFORMATION MOJETTE

Jury

Rapporteurs :	P. DUHAMEL	Directeur de Recherche CNRS, L2S, Gif Sur Yvette
	C. GUILLEMOT	Directeur de Recherche INRIA, IRISA, Rennes
Examinateurs :	D. BARBA	Professeur des Universités, EPUN/IRCCyN, Nantes
	JP. GUEDON	Maître de Conférences, EPUN/IRCCyN, Nantes
	N. NORMAND	Maître de Conférences, EPUN/IRCCyN, Nantes
	C. OLIVIER	Professeur des Universités, IRCOM-SIC, Poitiers

Directeur de thèse : Dominique BARBA Laboratoire : IRCCyN
Co-encadrant : Nicolas NORMAND Laboratoire : IRCCyN
Composante de rattachement du directeur de thèse : École polytechnique de l'université de Nantes

N° ED 0366-xxx

Table des matières

Table des figures **v**

Introduction **1**

Partie I État de l'art

Introduction de la première partie **7**

Chapitre 1
Introduction aux codages de source et de canal **9**

1.1 Codage de source - représentation graduable 10
1.1.1 Stratégie multi-résolution pour le codage d'images fixes 10
1.1.2 Méthodes graduables de codage vidéo 12
1.1.3 Robustesse aux pertes de paquets 15
1.2 Transmission d'une source graduable sur l'Internet 18
1.2.1 Vous avez dit réseaux à qualité de service ? 18
1.2.2 Modélisation du canal 24
1.2.3 Codage conjoint source/canal 26
1.3 Conclusion 27

Chapitre 2
Mécanismes de protection par description multiple de l'information **29**

2.1 Codes correcteurs par anticipation (FEC) 30
2.1.1 Les codes MDS 32
2.1.2 Un autre code proche de la solution optimale 36
2.2 Codage à description multiple 38
2.2.1 Le problème initial 38
2.2.2 Applications 40

2.2.3 Codages à description multiple 40
2.3 Transmission graduable multimédia 44
2.3.1 Protection graduable par description multiple 44
2.3.2 Protection graduable par codage FEC 45
2.3.3 Protocoles de transmission multimédia graduables 46
2.4 Conclusion 48

Chapitre 3
Représentation par transformation Mojette 51

3.1 Introduction 51
3.2 Opérateur Mojette-Dirac 52
3.2.1 Opérateur Mojette direct 52
3.2.2 Supports rectangulaires et condition de reconstruction [Katz 79] 54
3.2.3 Autres supports et théorème de reconstructibilité [Normand 97] 55
3.2.4 Opérateur Mojette inverse 56
3.2.5 Algorithmes et complexité 58
3.3 Utilisation en transmission réseaux 60
3.3.1 Redondance 61
3.3.2 Reconstruction inexacte 61
3.3.3 Approche multirésolution et transformation Mojette 64
3.4 Conclusion 67

Conclusion de la première partie 71

Partie II Système de protection inégale par représentation Mojette

Introduction de la deuxième partie 75

Chapitre 4
Protection Mojette non graduable 77

4.1 Choix du support et choix des projections 78
4.1.1 Protection Mojette 78
4.1.2 Propositions de mémoire tampon géométrique 79
4.2 Représentation Mojette et code MDS 91
4.2.1 Expression du coût supplémentaire par rapport aux codes MDS 91
4.2.2 Résultats 93

4.3 Conclusion 96

Chapitre 5
Protection à n niveaux de priorité 99

5.1 Codage prioritaire par transformée Mojette 100
5.2 Choix du support et choix des projections 101
5.2.1 Projections concaténées 102
5.2.2 Supports concaténés 104
5.3 Autre construction de mémoire tampon géométrique pour les supports concaténés 109
5.3.1 Optimisation de la capacité du support 109
5.3.2 Nombre d'ixels frontière 111
5.3.3 Construction finale 112
5.4 Application à la transmission d'image 116
5.5 Conclusion 117

Chapitre 6
Description multiple de l'information par transformation Mojette 119

6.1 Propriétés de description multiple 120
6.2 Optimisation du système de protection inégale 121
6.2.1 Allocation optimale de la redondance 121
6.2.2 À propos de la qualité 124
6.2.3 Voies de retour canal 133
6.3 Protection égale et protection inégale 135
6.3.1 Débit constant 135
6.3.2 Débit variable 138
6.3.3 Effet de pertes 139
6.4 Conclusion 141

Conclusion de la deuxième partie 143

Partie III Applications sur réseaux

Introduction de la troisième partie 147

Chapitre 7
Représentation multiple en transmission multimédia 149

7.1 Transmission d'images 150
7.1.1 Calcul de la fonction de priorité et de la mémoire tampon géométrique 150
7.1.2 Simulation de transmission avec pertes 152
7.2 Protocole de transfert ionosphérique 154
7.2.1 Conditions de transmission 155
7.2.2 Protocole de transmission 156
7.3 Description multiple à contraintes temporelles 156
7.4 Conclusion 158

Chapitre 8
Représentation multiple en stockage de données 159

8.1 Système d'informations distribuées 160
8.1.1 Exemples de systèmes distribués multimédia sur IP 160
8.1.2 Rôle de la pile RTP/UDP/IP et des relations avec le système Mojette 161
8.1.3 Redondance projection/paquet 162
8.2 Application de télémédecine 163
8.3 Conclusion 164

Conclusion de la troisième partie 167

Conclusion et perspectives

Annexes

Annexe A Mise en œuvre de la Mojette XOR 173

Annexe B Algorithme de la Mojette inverse 175

Bibliographie 179

Production scientifique 187

Table des figures

1.1 Script JPEG progressif : exemple d'un train binaire. 11
1.2 Schéma IPB utilisé dans les normes vidéo MPEG et H.26x 13
1.3 Rétablissement des données par VLC 16
1.4 Rétablissement des données par RVLC 17
1.5 Diagramme des 4-nœuds d'ARPAnet (1969). 18
1.6 Exploration partielle de l'Internet 2001 à partir du moniteur ihr.skitter.caida.org limitée à 535 102 nœuds. 19
1.7 Perte de paquets sur une semaine entre NOC (Los Angeles) et Brisbane (Australie). 22
1.8 Perte de paquets sur une journée entre NOC (Los Angeles) et Brisbane (Australie). 22
1.9 Perte de paquets sur 7 jours entre Los Angeles (CA) et Atlanta (GA). . . . 22
1.10 Perte de paquets sur une journée entre Los Angeles (CA) et Atlanta (GA). 23
1.11 Schéma général de transmission avec séparation du codage de source et du codage de canal. 24
1.12 Modèle de Gilbert-Elliott (R : paquet reçu ; P : paquet perdu). 26

2.1 Comment fonctionne un codage FEC ? A partir d'un ensemble de k paquets de message, un ensemble redondant de n paquets est codé ($n \geq k$). La réception d'un ensemble supérieur ou égal à k permet le décodage du message original. 32
2.2 Représentation en niveaux de gris d'une matrice de Cauchy 500×500 sur $GF[2^{10}]$. 34
2.3 Extraction de la sous-matrice carrée de Cauchy (5×5) pour un scénario de pertes de 5 paquets messages (d'indices : 11, 15, 43, 53, 68) compensées par 5 paquets redondants (d'indices : 4, 8, 9, 13, 22). 35
2.4 Temps de décodage en fonction du nombre de paquets redondants utilisés . 36
2.5 Structure du code Tornado. 37
2.6 Partage de l'information sur deux sources avec trois destinations. 39

2.7 (a) une quantification uniforme de 4 bits. Il n'y a aucune manière pour sectionner les quantificateurs en deux paires de bits pour obtenir une solution acceptable dans les décodeurs latéraux. (b) deux quantifications à 3 bits qui se complémentent pour donner une résolution de 4 bits au décodeur central. 41

2.8 Quantification scalaire à description multiple. Un exemple d'allocation d'index pour les coefficients quantifiés. 42

2.9 Autre exemple d'allocation d'index pour les coefficients quantifiés. 42

2.10 Transmission graduable multimédia. Le problème d'hétérogénéité peut être résolu en combinant le codage de source en couches avec une transmission graduable. Le nœud courant envoie seulement le nombre de couches que les liens peuvent supporter. 47

2.11 Réactivité temporelle du système RLM. Ce diagramme représente la tratégie de base d'adaptation pour un récepteur donné. Initialement, le récepteur démarre la couche de base puis graduellement ajoute de nouvelles couches jusqu'à la congestion (C). L'effacement d'une couche permet d'ajuster sa souscription à la source. 48

3.1 Transformation Mojette-Dirac sur une image 4×4 : projections (1,1) et (0,1). 53

3.2 Ouverture d'un carré par un élément structurant à deux pixels (ÉS2P). . . 56

3.3 Construction par dilatation de la plus petite image non reconstructible. . . 57

3.4 Exemple de transformation Mojette inverse : support 4×4 reconstruit à partir des trois projections {(1,1) ;(-1,1) ;(-2,1)}. 58

3.5 ÉS2P valué pour S_3 = {(1,0),(1,1),(2,1)}. 62

3.6 Fantôme élémentaire pour S_3 = {(1,0),(-1,1), (2,1)}. 62

3.7 Région érodée par les projections (-1,1) et (1,0). 63

3.8 Décomposition unique par les fantômes de base issus des projections (-1,1) et (1,0). 64

3.9 Reconstruction de l'image Lena et image différence après $j = 1$ filtrage de Haar sur chaque projection de l'ensemble $S = \{(16, 1); (8, 1); (4, 1); (-4, 1)\}$. 65

3.10 Analyse ondelette (W : Wavelet) d'une image et transformation Mojette (TM) sur chaque bande de fréquences spatiales. 65

3.11 Transmission multi-résolution et multi-canaux. 67

3.12 Reconstruction asynchrone par la transformation Mojette inverse. L'algorithme peut débuter à l'arrivée de la première projection (-1,1) en initialisant la valeur des bins des autres projections (1,0) et (-2,1) à zéros. La contribution des éléments reconstruits est retiré des projections que celles-ci soient ou ne soient pas reçues. 69

4.1 Représentation des vecteurs de projections (résultante en pointillé) sur un support rectangulaire avec $Q = 4$ pour trois sous-ensembles reconstructibles parmi l'ensemble $S = \{(-2,1),(-1,1),(0,1),(1,2),(2,1)\}$. (1) : $\{(-2,1),(-1,1),(0,1),(2,1)\}$; (2) : $\{(-1,1),(0,1),(1,2)\}$; (3) : $\{(0,1),(1,2),(2,1)\}$. 79
4.2 Comportement de la redondance pour q = {1, 5, 11, 27} en fonction de la capacité d'une mémoire tampon géométrique rectangulaire. 82
4.3 Représentation des parcours sur deux supports distincts pour la projection (1,2). 83
4.4 (a) Exemple d'une mémoire tampon géométrique rectangulaire 6x3. (b) chaque élément de la mémoire est exactement reconstructible par trois projections prises dans l'ensemble S. 84
4.5 (a) Exemple d'une mémoire tampon géométrique hexagonal. (b) Chaque élément de la mémoire est reconstructible par trois projections prises dans l'ensemble S. (c) les 10 dilatations des ES2P correspondant aux 10 combinaisons de recevoir 3 projections parmi 6. 86
4.6 La reconstruction du support par M projections implique une épaisseur de M ixels. 87
4.7 Construction de l'extremité gauche du support hexagonal. L'ixel X1 est dans le premier bin de toutes les projections. 88
4.8 Forme de la mémoire tampon géométrique hexagonal. 89
4.9 Points intérieurs et calcul de surface d'un polytope. 90
4.10 Différence de capacité entre la mémoire tampon hexagonale et la mémoire tampon rectangulaire. 92

5.1 Exemple de la projection d'une mémoire tampon géométrique avec trois niveaux de protections. La réception de trois projections (resp. quatre et cinq) permet la reconstruction de la région 1 (resp. 2 et 3). 101
5.2 Schéma de protection à trois niveaux de priorité avec séparation des mémoires tampon géométriques. La mise en paquets pour un angle donné s'effectue par concaténation des projections de chaque mémoire. La largeur de la mémoire est fonction de la taille des paquets (contrainte canal). 103
5.3 Schéma de protection à trois niveaux de priorité avec concaténation des mémoires tampon géométriques. Il y a dans ce cas correspondance entre un paquet et une projection. Les unités de transport ainsi crées sont de taille constante. 106
5.4 Exemple de frontières entre deux niveaux de protection pour deux scénarios de réception. La dilatation des éléments structurants pour les deux scénarios délimite les deux régions. 107
5.5 Arrêt de la reconstruction de la zone protégée m_1 par manque de correspondance univoque. 108

5.6 Optimisation de la capacité du support. L'ajout des deux ixels sur le support 2 par rapport au support 1 ne modifie pas les possibilités de reconstruction (reconstruction totale par cinq projections). 110
5.7 Détails du comportement à la reconstruction des ixels situés à la frontière de deux régions. La région 1 est reconstructible par trois projections et la région 2 est reconstructible par sept projections. 112
5.8 Concaténation des supports par rapport à un axe. La région de transition est à repréciser. 113
5.9 Résultat de la réunion des deux ensembles d'ixels frontière non reconstructibles par les deux sous-ensembles de trois projections {(-3,1) ; (-2,1) ; (-1,1)} et {(3,1) ; (2,1) ; (1,1)}. 113
5.10 Nombres d'ixels reconstruits en fonction du nombre de projection reçues et de leur ordre d'arrivée. 115
5.11 Multiplexage de trois sous-flux JPEG sur une mémoire tampon géométrique à trois niveaux de protection. 116
5.12 Résultats de la protection Mojette pour l'image Rose codée en JPEG progressif à 1,33 bpp. De gauche à droite, le nombre de paquets perdu est de 3, 2 et 1 parmi 6 pour un débit de 0,45 bpp, 0,79 bpp et 1,33 bpp. 117
5.13 Présentation d'une mémoire tampon géométrique contenant trois niveaux de protections. La région de gauche (resp. centrale et de droite) est reconstructible par trois (resp. cinq et sept) projections parmi sept. 118

6.1 Répartition des flux dans N paquets. 123
6.2 Images Léna dégradées comportant un même PSNR de 32,09 dB. L'image de gauche comprend une dégradation très localisée alors que l'image de droite est issue d'un codeur SPIHT (0,158 bpp). 125
6.3 Décodage du flux 1 et reconstruction de l'image bateau 512×512 (débit 0,4 bits/pixels pour un PSNR de 32,22 dB). 126
6.4 Décodage du flux 2 et reconstruction de l'image bateau 512×512 (débit 0,8 bits/pixels pour un PSNR de 37,02). 127
6.5 Image requise par le service. L'acceptabilité de service se définit ici simplement par un niveau de qualité de 34,25 dB (0,55 bits/pixel). 128
6.6 Calcul de l'espérance mathématique de la qualité pour les couples $(m1, m2)$ et pour un profil de perte de paquet exponentiel de valeur moyenne 10 %. . 129
6.7 Résultat de l'intersection du seuil de qualité acceptable avec la courbe d'espérance mathématique de la qualité. 130
6.8 Niveaux de protection m_1 et m_2 pour les deux sous-flux source (1 et 2) pour un niveau de qualité acceptable de 36 dB. 131
6.9 Niveaux de protection m_1 et m_2 pour les deux sous-flux source (1 et 2) pour un niveau de qualité acceptable de 34 dB. 132

6.10 Niveaux de protection m_1 et m_2 pour les deux sous-flux source (1 et 2) pour un niveau de qualité acceptable de 32 dB. 132

6.11 Niveaux de protection m_1 et m_2 pour les deux sous-flux source (1 et 2) pour un niveau de qualité acceptable de 29 dB. 133

6.12 Débit d'une source TCP et de deux sources TCP-friendly pour une connection entre Toronto et Berkeley [Tan 99]. 135

6.13 Mesure de la qualité en fonction du nombre de descriptions reçues pour une protection égale (EEP) et pour une protection inégale (UEP) non optimisée. 137

6.14 Mesure de la qualité en fonction du nombre de descriptions reçues pour une protection égale (EEP) et pour la protection inégale (UEP) optimisée. . . . 137

6.15 Mesure de la qualité espérée à la réception en fonction du nombre de paquets envoyés pour un taux de perte moyen de 10 %. L'obtention de l'image de la figure 6.5 correspond au seuil de qualité (en pointillés sur la courbe). . . 138

6.16 Mesure de la qualité espérée à la réception en fonction du nombre de paquets envoyés pour un taux de perte moyen de 20 %. 139

6.17 Comparaison des reconstructions entre la description multiple (DM) et le codage SPIHT avec retransmission. 140

7.1 IRM abdominale 256×256 en 16 bits par pixel. 150

7.2 Profil exponentiel de pertes pour une valeur moyenne de 10 %. 151

7.3 Buffer géométrique construit pour la fonction de priorité indiquant que 11 projections sont nécessaires pour reconstruire le flux 1 d'une image SPIHT 256×256 ; 14, 15 et 16 pour les flux 2, 3, 4 respectivement. 152

7.4 Mesures du PSNR en fonction du nombre de projections reçues. 152

7.5 Mesure de redondance à la réception en fonction du nombre de projections reçues. 153

7.6 IRM abdominales codés avec SPIHT. (a) débit = 0,255 bpp avec onze projections nécessaires. (b) débit = 0,535 bpp avec quatorze projections nécessaires. (c) débit = 1,049 bpp avec quinze projections nécessaires. (d) débit = 1,245 bpp avec seize projections nécessaires. 154

7.7 Types de trajets ionosphériques possibles. 155

7.8 Représentation de la solution hybride de protection combinant les projections (n-1)-D et (n-2)-D. 157

8.1 Système distribué Mojette composé de trois serveurs et six clients. 161

8.2 Accès et stockage à une base de données images médicales distribuée sur deux sites. 164

A.1 Mise en œuvre de la transformation Mojette XOR au travers du calcul des projections de type (0,1), (1,1) et (-1,1) sur un support binaire d'information de dimension 4×4. 174

Remerciements

Tout d'abord, mes remerciements s'adressent aux rapporteurs, les Professeurs Guillemot et Duhamel, qui ont accepté la lecture attentive de ce document malgré un emploi du temps souvent chargé.

Je remercie les membres du jury et Dominique pour leurs critiques, encouragements et enthousiasme de chercheurs.

Quand je me suis présenté au laboratoire IVC muni d'une question qui me tarabustait dans mon stage de DEA, Nicolas Normand après avoir fini d'y répondre me proposa de but en blanc :
"Ça te dirait de faire une thèse ?"
Après une série de termes abscons, j'exprimais un doute sur cette orientation qu'il refoula par un magnifique :
"Tu n'aimes pas l'aventure ?"
Ses propos simples, précis er justes m'ont guidé pendant ces trois ans.

Autre compagnon de route : Jean Pierre Guédon, enthousiaste de la première heure. Je salue, ici, ses idées, son attention et ses Mojettes théoriques et consistantes.

Je remercie mes parents, ma soeur, pour leur soutien sans faille, les Tomy's, Jean Val, potes de toujours, Fred, Vio et leurs jumeaux fraîchement débarqués, Steph et Nico, Michou et Mich'pat et leurs habits de lumière, Nico, Nono et Vigneux-de-Bretagne, Professeur Bizais et M'sieur Couture pour avoir dopé ma curiosité, Vivek Goyal et notre rencontre aux Bell Labs, les potes de l'Inserm, Sigrid, Jeff, le Calyp's atlantic, Moktar et son steel band, ma batterie,

les membres exotiques de l'équipe IVC de Vigo à Kuala Lumpur, de la ville du Cap à Sköpje et plus particulièrement Pedro, Freddy, John, Florent, Pierre, Bwana, Big Sim, gâ Pouliq', Ali, Lionel, le petit scarabée et Amal, le rayon de soleil du laboratoire,

et enfin, Sarah, pour avoir supporté mes absences, pour les retrouvailles, son sourire et son écoute.

à Sarah,

Introduction

Le problème fondamental de la communication reste le même depuis une cinquantaine d'année : la reproduction en un point, soit exactement soit approximativement, d'un message envoyé à partir d'un autre point. En prévision, il a été avancé récemment que pour les cinquante années à venir les télécommunications seront dominées par la transmission par paquets de la voix, des données sur des réseaux mondiaux tels que l'Internet. Le défi sera de maitriser le réseau dans sa globalité et de garantir une qualité de service de bout en bout [Calderbank 98]. C'est dans ce contexte d'étude que s'inscrivent les travaux de cette thèse et plus particulièrement la transmission de données multimédias sur l'Internet public, réseau à qualité de service non garantie.

L'existant

L'approche classique d'une transmission est décomposable en deux étapes.
L'une a pour objectif de réduire fortement le débit de données pour pouvoir les envoyer. On parle alors de **codage de source**. Sa fonction est de supprimer les éléments redondants de la source au prix d'une baisse de la qualité de représentation à des taux de compression élevés. Son efficacité est fortement liée aux dépendances statistiques contenues dans la source et surtout aussi à la nature de la représentation de l'information initiale.
L'autre étape constitue un mécanisme de protection contre les erreurs de transmission. Elle concerne le **codage de canal.** À l'issue de ce codage, une partie du débit transmis est consacrée à des données de redondance pour réduire la probabilité d'erreurs résiduelles. L'information contextuelle de la source qu'elle soit d'image, de vidéo ou de voix est absente de sa mise en œuvre.
D'après le principe de séparation de Shannon, le codage de source et le codage de canal peuvent être optimisés de manière indépendante. Ce modèle de transmission est générique et s'applique à une grande diversité de sources.
Un problème, cependant, se pose quant aux niveaux de pertinence de l'information véhiculée. L'absence de contexte ne permet pas de distinguer l'information importante de celle superflue. L'optimisation conjointe des codages de source et de canal constitue donc une alternative à l'approche classique.
Le domaine de recherche du **codage conjoint source-canal** est aujourd'hui en vogue

compte tenu de l'évolution des applications dites multimédias et de la part sans cesse accrue des communications mobiles. Cependant, nous verrons que les solutions généralement proposées pour résoudre le problème fondamental de la communication demeurent proches de l'une ou de l'autre problématique justifiées par l'importante contribution de Claude Shannon.

Les besoins en termes d'efficacité du codage de source sont dues aux limites de capacité du canal. Aujourd'hui, une portion significative de l'Internet met en jeu des connections via la ligne téléphonique pour des débits de 56 kbit/s dans les meilleures configurations. En regard des évolutions permanentes de la bande passante disponible (modem câble, technologies xDSL[1]) et dans une perspective de recherche, il paraît raisonnable de s'affranchir de la contrainte des "derniers kilomètres" (boucle locale).
Les applications actuelles nécessitent de mettre en œuvre des transformations permettant d'obtenir plusieurs niveaux de résolutions (représentation hiérarchique ou graduable[2]). En revanche, la structure du support de communication, en l'occurrence l'Internet, impose de concevoir des représentations de la source robustes aux pertes aléatoires de paquets d'information.
Un hiatus se dégage de cette situation entre la représentation hiérarchisée et très structurée des sources actuelles et l'absence de traitements des priorités sur le canal de communication.

Notre approche

Dans le cadre de transmission sans qualité de service garantie, les transformations à description multiple apportent des possibilités pour composer un mécanisme de compensation de pertes de paquets. Les descriptions partielles générées sont en effet équivalentes et peuvent être effacées de manière arbitraire.
Intuitivement, le codage à description multiple constitue une autre approche du raffinement successif de la source, où celle-ci (*e.g.* une image) peut être divisée en deux descriptions p_1, p_2. La différence est que l'on suppose que p_1 soit reçue pour pouvoir utiliser la description p_2 qui, dans ce cas, augmente strictement le critère de fidélité à la réception. Si p_1 est perdue p_2 ne peut pas être décodée même si elle arrive intacte. Pour prendre un exemple, le décodage d'une source JPEG progressif ne peut pas être réalisé si seulement les dernières portions du flux sont reçues. Dans le contexte de la description multiple, quelles que soient les descriptions reçues p_1 ou p_2, le décodage est à même de fournir une version dégradée dans les deux cas et d'augmenter le critère de fidélité dans le cas favorable de la réception des deux descriptions.
Nous utiliserons pour cette approche un outil simple et efficace de description d'infor-

[1] Digital Subscriber Loop
[2] Plusieurs traductions de l'anglais sont disponibles pour le terme "scalability", nous retenons dans cette thèse les termes "graduabilité" et "graduable".

mation : la transformation Mojette. Elle réalise à la base le passage d'un signal à deux dimensions vers plusieurs signaux à une dimension. L'information est projetée géométriquement dans des directions particulières (directions formées par des couples d'entiers premiers entre eux) et fournit un ensemble de descripteurs appelées **projections.** Chaque projection constitue une unité de transport.
La redondance introduite par cette transformation justifie son utilisation dans un environnement à perte de paquets. En choisissant les angles ainsi que la forme du support d'information (on montre que le support naturel d'une image n'est pas approprié), la reconstruction est possible par n'importe quel sous-ensemble de projections constitué bien sûr d'un nombre suffisant de projections.

Contributions

Ce travail propose une nouvelle méthode de description multiple en utilisant la transformation Mojette. Avec un choix adéquat, détaillé dans cette thèse, du support d'information et des angles de projections et par la faible complexité de cette transformation, elle effectue un bon compromis entre contrainte d'intégrité et contrainte de temps réel pour le transport de sources hiérarchisées. Le nombre de projections générées permet en outre de généraliser le problème de la description multiple initialement formulé pour deux canaux à un ensemble de N canaux. Dans ce travail, le modèle du canal prend en compte en plus des caractéristiques de la transmission, tout ou partie des organes d'émission et de réception. On étend donc la définition du canal en englobant dans ce terme les aspects protocolaires inhérents à la transmission. Notre codage concerne ainsi principalement la transmission de sources graduables réelles sur le protocole IP, protocole de transport de l'Internet. Il ne freine pas l'évolution des sources actuelles et suppose un comportement "best effort" du canal. Au terme du codage, les unités de transport sont banalisées, ne nécessitent pas de traitements prioritaires et peuvent être perdues arbitrairement. Le protocole de transport des projections est ainsi relativement simplifié.

La part importante est donc donnée à la construction des supports d'informations et sur le choix des angles de projections pour un flux de données de même priorité et pour un ensemble de sous-flux de priorités distinctes. La préparation des données source structurées avant l'émission confère à ces supports des propriétés similaires aux mémoires tampon classiques. Dans notre contexte, on parlera donc indifféremment de supports d'information ou de mémoires tampon géométriques. La projection multiple apporte cependant des fonctionnalités supplémentaires comme le contrôle d'intégrité. La redondance introduite par la nouvelle description de la source se caractérise par un nombre N de projections. Comme il est précisé avant, toutes ces projections ont même pouvoir de reconstruction et donc un sous-ensemble de M projections parmi N, quelles qu'elles soient, suffisent à reconstruire le support. Cette propriété est celle des codes correcteurs d'erreurs MDS (*Maximum Distance Separable*), mais avec des symboles messages de même taille que les

symboles redondants. Dans le cas de la représentation Mojette, le nombre de symboles qui constituent un ensemble de projections reconstructible peut être légèrement supérieur au nombre des symboles de message. Il faut en effet un peu plus d'éléments qui constituent une projection (éléments appelés bins) que de symboles de la source initiale pour parvenir à l'inversion et la reconstruction du support. Le code formé revêt alors les propriétés d'un code $(1+\varepsilon)$**MDS** mais pour une complexité très réduite. Les paramètres qui influent sur ce surplus d'informations nécessaires au décodage sont précisés et une solution proche du cas optimal MDS est proposée.

Enfin, nous effectuons une optimisation du système de protection de la hiérarchie de la source en allouant les nombres de projections nécessaires pour chaque sous-flux source en maximisant un critère de qualité. Cette opération utilise de manière combinée les informations de la source et du canal de transmission.

Organisation du document

Ce document est organisé en trois parties.

Nous proposons tout d'abord un état de l'art concernant la transmission d'une source hiérarchisée sur un canal de l'Internet public. Le chapitre 1 introduit les notions de codage de source et de codage de canal au travers d'exemples de mises en œuvre actuelles. Observant les limites d'une approche séparée pour la transmission robuste d'information, les solutions de codage à description multiple sont référencées dans le chapitre 2 incluant l'approche des codes correcteurs par anticipation (*Forward Error Correcting codes*). Le chapitre 3 fournit les résultats fondamentaux relatifs à la transformation Mojette.

La deuxième partie décrit, dans un premier temps, la construction des mémoires tampons géométriques pour la protection égale par transformation Mojette (chapitre 4) qui délivre un seul niveau de priorité. La propriété $(1+\varepsilon)$MDS du code est en outre étudiée. Le chapitre 5 présente la protection inégale pour la transmission d'une source graduable. Dans le chapitre 6, on compare les deux approches des deux chapitres précédents et on énonce les propriétés du codeur à description multiple de l'information par transformation Mojette.

Dans une troisième partie, nous présentons les applications du système dans le domaine de la transmission (chapitre 7) et dans le domaine du stockage de données (chapitre 8). Enfin, nous terminerons ce document par une conclusion et les perspectives de ce travail.

Première partie

État de l'art

Introduction de la première partie

Le transfert de données numériques est prioritairement conditionné par des contraintes d'intégrité. L'objectif est de faire en sorte que les données issues de la source d'information soient transmises de façon sûre à travers le canal de transmission.
Nous allons dans cette première partie tracer l'état de la recherche dans le domaine de la transmission d'une source de type image. Cet état de l'art ne prétend évidemment pas être une description exhaustive de tous les types de méthodes de transmission car ce travail fait bel et bien appel à une recherche dans deux communautés distinctes ayant chacune ses propres méthodes, problématiques et conférences : le codage de source (traitement du signal) et le codage de canal (réseaux et télécommunications). Nous tenterons d'établir une vue générale de ces approches afin d'en délimiter les faiblesses.

Les différents besoins en termes de communication et de services multimédias sont en partie satisfaits par l'existence de grandes artères communément appelées autoroutes de l'information capables de véhiculer des quantités d'information de plus en plus importantes, par le biais de réseaux à haut débit (e.g. gigabit Ethernet) avec des protocoles de transmission par paquets tels que l'*Internet Protocol* (IP). Cette course à la bande passante s'illustre par des performances en perpetuelle évolution (un terabit/s sur une distance de mille kilomètres avec les techniques combinées de solitons et DWDM[3]). Malgré l'augmentation de la capacité du canal, le contrôle d'intégrité demeure une fonctionnalité indispensable placée aux extrémités des liaisons de communication. Ce contrôle n'étant pas effectué sur le trajet, des dégradations et des pertes de paquets d'information peuvent survenir pendant la transmission. Le message délivré est alors incomplet.

Dans un premier temps, nous évoquerons séparément le codage de source et le codage de canal et les méthodes s'y référant pour la transmission robuste d'information. Le codage d'image passe le plus généralement par un changement d'espace de représentation via des transformations. On passe classiquement d'une représentation spatiale interprétable par l'œil humain à une autre représentation pour mieux exploiter les dépendances statistiques et ainsi réduire la quantité d'information. Ces représentations sont très courantes et encore largement exploitées dans les codeurs dits de deuxième génération dont la finalité est la description du signal en tant qu'objet (JPEG-2000, MPEG-4). La sortie de ces codeurs

[3]Dense Wavelength Division Multiplexing

fournit des informations très structurées (structure graduable) qui entraîne, nous le verrons, la définition de priorités dans la transmission sur un canal bruité. Cette structure présente néanmoins des faiblesses dans sa réalisation. La perte d'une portion du flux peut compromettre en effet le décodage de l'intégralité de l'information transmise issue de la source codée. Une étape supplémentaire consiste alors à inclure uniformément de la redondance lors du codage de canal. Dans le cadre du codage conjoint source-canal, nous montrerons comment adapter les besoins de la source aux caractéristiques du canal (délai de transmission, taux de pertes, qualité de service sous-jacente...).

C'est ainsi que dans un deuxième temps, les mécanismes de protection par description multiple seront présentés. En cassant la structure multirésolution, nous exposerons le moyen de représenter la source par un ensemble de descripteurs d'importance équivalente visant à compenser efficacement les pertes aléatoires sur le canal de communication. La propriété d'équivalence est fortement requise sur des canaux de type "best effort" issues de la simplification des protocoles. Nous ferons un état des applications actuelles sur ce type de canal.

Enfin, dans un dernier temps, au chapitre 3, nous présenterons une transformation réalisant simplement un ensemble de descriptions. Nous en détaillerons les principales propriétés et montrerons qu'elle est capable de concilier les contraintes de temps-réel avec les aspects de réaction aux pertes d'information.

Chapitre 1

Introduction aux codages de source et de canal

Sommaire

1.1 Codage de source - représentation graduable 10

1.1.1 Stratégie multi-résolution pour le codage d'images fixes . . . 10

1.1.2 Méthodes graduables de codage vidéo 12

1.1.3 Robustesse aux pertes de paquets 15

1.2 Transmission d'une source graduable sur l'Internet 18

1.2.1 Vous avez dit réseaux à qualité de service ? 18

1.2.2 Modélisation du canal 24

1.2.3 Codage conjoint source/canal 26

1.3 Conclusion 27

Pour transmettre de façon fiable de l'information issue d'une source à une destinanataire, Claude Shannon suggère une méthode de transmission se découpant en deux procédés successifs :

- un codage de source, la compression, générant un débit devant être inférieur à la capacité du canal ;
- un codage de canal, la protection, utilisé pour transmettre la source au destinataire avec capacité de correction des erreurs de transmission éventuelles.

Ce découpage est la conséquence du théorème de codage source-canal stipulant qu'il est possible d'atteindre une distorsion minimale à la réception par une optimisation indépendante des deux blocs : le codage de source et le codage de canal. Ce principe est la base des communications numériques actuelles car il permet la mise en œuvre de techniques de compression indépendamment du médium utilisé. Le canal peut en effet se décliner sous de nombreuses formes. De ce panel de canaux, nous pouvons citer les unités de stockage classiques comme le CD-ROM ou les disques RAID (*Redundant Arrays of Inexpensive Disks*)

ainsi que les supports de transmission hertziens ou câblés. Pour ce dernier type, le mode de transmission par paquets s'impose aujourd'hui en communication quel que soit le niveau physique du canal. Mais avec quelle fiabilité les symboles de communications peuvent-ils être transmis ? Ce problème technique que décrivait Warren Weaver [Weaver 49] est encore présent dans ce mode de transmission sujet à l'effacement de paquets. En outre, avec quelle précision les symboles transmis véhiculent-ils la signification désirée ? De ce problème sémantique dépend la robustesse de la source aux pertes de paquets ou sa faculté à discriminer l'information importante de celle superflue (représentation graduable).
Aussi ce premier chapitre aborde-t-il de manière classique la chaîne de transmission. La première partie est consacrée aux codages de sources image et vidéo. Au travers de codes existants, après leur présentation, on étudiera les mécanismes de compensation de paquets mis en œuvre. La deuxième partie est dédiée au canal et particulièrement à l'Internet public, canal à effacement par excellence. Des éléments de la théorie de l'information sont donnés pour une modélisation de ce canal.

1.1 Codage de source - représentation graduable

Cette section aborde la représentation graduable dans les sources actuelles d'image et de vidéo que l'on rencontre plus ou moins classiquement sur l'Internet. Elle ne dresse en aucun cas un état de l'art complet sur les méthodes de compression mais s'applique à la description des composants d'un train binaire source en vue d'une séparation en sous-flux à protéger distinctement.

1.1.1 Stratégie multi-résolution pour le codage d'images fixes

Le codage de source pour l'image met l'accent sur la progressivité du train binaire obtenu. Cette représentation multi-résolution ou en couche (LC[4]) offre la possibilité au décodeur de reconstruire d'emblée l'image originale de manière grossière sans attendre la totalité de l'information. L'élaboration de la hiérarchie de la source est naturelle dans une décorrélation par transformation orthogonale. La transformation en cosinus discrète (*Discrete Cosine Transform : DCT*) est adoptée pour le codage JPEG progressif. En revanche, une transformation en ondelettes (*Discrete Wavelet Transform*) est utilisée dans les méthodes de codage plus récents avec progressivité comme SPIHT[5] ou EBCOT[6] (JPEG-2000). Ces trois codeurs sont succinctement présentés dans la suite de cette section.

[4] Layered Coding
[5] Set Partitionning in Hierarchical Trees
[6] Embedded Block Coding with Optimized Truncation

JPEG Progressif

La norme JPEG (*Joint Photographic Experts Group*) est un codage d'image avec compression d'information basée sur la DCT. Un découpage du support de l'image en blocs de taille fixe (8×8 ou 16×16) est préalablement effectué avant d'appliquer la DCT. Dans le mode de codage avec perte, la compression est réalisée lors de l'étape de quantification. Les éléments issus de la DCT sont organisés selon une séquence 1-D par un balayage zig-zag à partir de l'élément a_{00} représentant la composante continue (élément DC). Le résultat de la DCT concentre l'énergie du signal de départ sur quelques éléments liés aux basses fréquences. Les éléments considérés comme moins importants (caractérisant les moyennes et hautes fréquences de l'image noté AC) ont en général des valeurs plus faibles et peuvent donc être quantifiés grossièrement même éventuellement supprimés. La norme JPEG utilise cette transformation DCT ainsi qu'une loi de quantification uniforme propre à chaque élément et un codage entropique des éléments quantifiés.
Une version progressive est mise en œuvre pour satisfaire la contrainte de graduabilité au train binaire final. Ce codage permet d'entrelacer les coefficients DCT quantifiés des composantes achromatiques et chromatiques avec sélection spectrale. Au fur et à mesure de l'arrivée du train binaire, l'ensemble des coefficients retrouvent leur quantification initiale de manière progressive. Le script de la figure 1.1 est un exemple de formation du train binaire JPEG progressif pour des images YCbCr offrant une approximation successive de la source [Group 98]. Les composantes Y, Cb et Cr sont respectivement étiquetées par 0, 1, 2 et suivies de trois paramètres : les indices des coefficients, les bits transmis et non transmis.

```
# Coefficients DC pour Y, Cb, Cr (le bit de poids faible n'est pas envoyé)
      0,1,2: 0-0,   0, 1 ;
# 5 premiers coefficients AC pour Y (moins les deux derniers bits)
      0:     1-5,   0, 2 ;
# Tous les coefficients AC pour Cr, Cb (moins le bit de poids faible)
      2:     1-63,  0, 1 ;
      1:     1-63,  0, 1 ;
# Coefficients AC restants pour Y (moins les deux derniers bits)
      0:     6-63,  0, 2 ;
# L'avant dernier bit de tous les coefficients AC pour Y
      0:     1-63,  2, 1 ;
# Bits de poids faible des coefficients DC pour Y, Cb, Cr
      0,1,2: 0-0,   1, 0 ;
# Bits de poids faible des coefficients AC pour Cb, Cr
      2:     1-63,  1, 0 ;
      1:     1-63,  1, 0 ;
# Bits de poids faible des coefficients AC pour Y; c'est habituellement la
séquence la plus longue
      0:     1-63,  1, 0 ;
```

Fig. 1.1: Script JPEG progressif : exemple d'un train binaire.

Codage SPIHT

La transformation en ondelettes est assimilable à l'analyse du signal image par un banc de filtres 2D, produit séparable de bancs de filtres 1D à deux bandes [Mallat 89] et un sous échantillonnage. La sortie du banc est constituée de quatre signaux : une sous-bande passe-bas LL et trois sous-bandes passe-haut HH, HL et LH. L'analyse peut être répétée de manière récursive sur la composante LL obtenue. Au moment de la synthèse, l'ajout de sous-bandes de fréquences supérieures augmente la résolution de l'image reconstruite.
La méthode SPIHT [Said 96a] (amélioration de la méthode de J. M. Shapiro (EZW[7])) exploite les dépendances statistiques existantes entre les coefficients de même orientation et de même localisation spatiale, mais appartenants à des sous-bandes de résolution différente. L'hypothèse de décroissance du spectre utilisée dans SPIHT indique que si un coefficient ondelette d'une résolution grossière est insignifiant pour un seuil donné, alors tous les coefficients de même orientation dans la même zone spatiale sont susceptibles d'être insignifiants vis-à-vis de ce seuil. Une structure arborescente différente des coefficients ondelettes permet de dépasser les performances d'EZW.

EBCOT

L'approche EBCOT fournit la représentation graduable adoptée par le standard de codage d'image fixe JPEG-2000 [JPEG2000] en cours de normalisation. Elle est basée sur une décomposition en ondelettes discrètes qui peut être issue indifféremment de la représentation multirésolution de Mallat, ou d'une décomposition quelconque en paquets d'ondelettes.
Chaque sous-bande est partitionnée en blocs d'échantillons B_i, de taille 32×32 ou 64×64. EBCOT génère pour chaque bloc un train binaire indépendant. Outre la graduabilité entre les flux binaires, chaque train peut être tronqué à des points R_i^n, correspondant à une distorsion D_i^n. La propriété intéressante de cet ensemble de points de troncature (R_i^n, D_i^n) est que la plupart de ces points sont situés sur la courbe débit-distorsion du bloc considéré. Ces fonctions inter et intra flux confèrent au codage une haute granularité dans la représentation de la source.
Afin d'organiser en couches le train binaire final, chaque contribution des blocs est déterminée par un algorithme d'optimisation débit-distorsion en fonction du débit alloué à chaque couche [Ortega 98]. Des méta-données s'ajoutent au flux final pour décrire les points de troncature et les longueurs associées à chaque bloc.

1.1.2 Méthodes graduables de codage vidéo

Le principe du codage comprimé de séquences d'images est la réduction de la redondance temporelle par estimation et compensation de mouvement. La plupart des standards

[7] Embedded Zerotree Wavelet

de compression vidéo adopte le schéma IPB décrit dans ce qui suit. On verra aussi dans ce paragraphe les spécificités des standards H263 et MPEG4.

Le schéma IPB

Ce schéma est utilisé dans les normes MPEG-1 et MPEG-2 : tel quel, ce ne sont pas (sauf MPEG-2 dans des profiles particuliers) des schémas graduables. Pour atteindre des taux de compression élevés, les redondances spatiales et temporelles contenues dans le signal d'image doivent être exploitées. Pour cela, les schémas de compression vidéo standard définissent trois modes de compression d'images :

- **image de type I (ou intra) :** véritable ossature du schéma, l'image intra est codée indépendamment des autres images de la séquence en réduisant la redondance spatiale contenue dans le signal.
- **image de type P (ou prédite) :** le codage de cette image exploite la redondance temporelle qui existe entre elle-même et l'image I ou P qui la précède et lui est la plus proche temporellement.
- **image de type B (ou bidirectionnelle) :** là aussi c'est la redondance temporelle qui est exploitée entre les images I ou P **précédentes** et **suivantes** pour fournir le plus faible débit des trois types d'images.

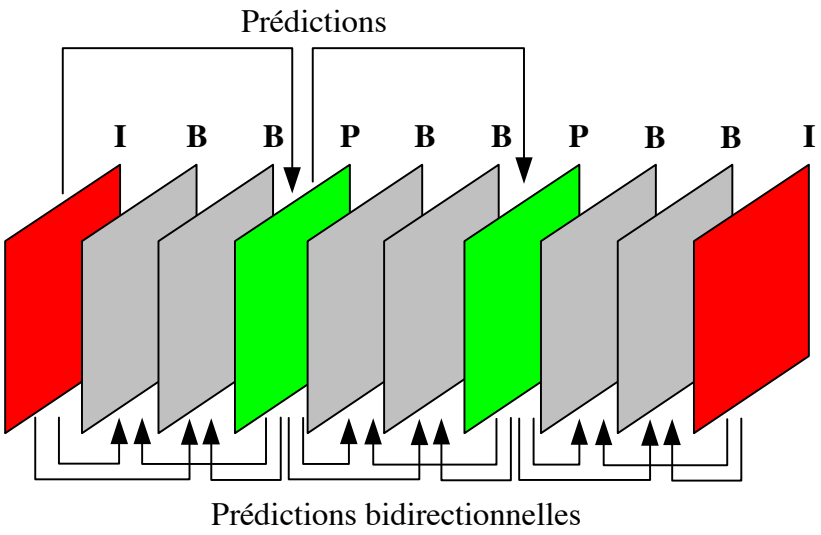


Fig. 1.2: Schéma IPB utilisé dans les normes vidéo MPEG et H.26x

Les images de type I sont codées de façon analogue au codage JPEG. Ces images codées peuvent servir ensuite de références aux images prédites P par estimation et compensation de mouvement de blocs. Pour chaque bloc de l'image courante à coder, un bloc de référence est selectionné dans une zone de recherche par minimisation d'un critère de type

similitude ou dérivée. A l'issue de cette sélection, le vecteur de mouvement correspondant au déplacement du bloc courant à coder par rapport au bloc contenu dans l'image de référence est déterminé. Ce vecteur de mouvement ainsi que l'erreur résiduelle entre le bloc prédit et le bloc de référence est transmis au décodeur. L'erreur résiduelle est elle-même codée comme le sont les images I mais avec des lois de quantification plus grossières que celles utilisés en mode I. Les images de type B sont codées de manière similaire avec deux images de référence respectivement antérieures et postérieures.

Standards H.263 et H.263+

La norme H.263 améliore les taux de compression observés dans H.261 qui est initialement prévue pour l'application en visio-conférence. Les images codées sont de meilleure qualité, les débits sont plus faibles et l'augmentation de la complexité reste raisonnable grâce aux graduabilités de qualité, temporelles et spatiales qu'offrent ce standard. La résistance du flux vidéo aux environnements à pertes de paquets est par la suite améliorée dans son extension H.263+.
La syntaxe vidéo H.263 se compose d'une hiérarchie en quatre couches : image, groupe de blocs (GOB pour *Group of Bloc*), macrobloc et bloc. Chaque image est précédée d'un en-tête comprenant les informations relatives à son mode de codage, comme le type de l'image (I, P ou B). Une image est structurée en groupe de blocs chacun structuré en un nombre entier de macroblocs. Chaque macrobloc correspond à un ensemble de quatre blocs de 8×8 pixels dans l'image fournissant quatre blocs d'échantillons de la composante achromatique et deux blocs d'échantillons de la composante chromatique (format 4 :2 :0). Selon le § 1.1.2, les vecteurs de mouvements sont estimés au niveau des macroblocs avec plusieurs modes disponibles.

- **mode I** : macrobloc codé en mode intra ;
- **mode P** : macrobloc prédit à partir d'une seule image de référence et un seul vecteur de mouvement ;
- **mode P4** : idem qu'en mode P mais macrobloc prédit par quatre vecteurs de mouvement soit un par bloc de luminance ;
- **mode U** : macrobloc non codé.

Les images B ne font pas partie du codage vidéo en mono-couche mais constituent les couches de raffinement en scalabilité temporelle.

MPEG-4

L'objectif du standard MPEG-4 est d'offrir à l'utilisateur de nouvelles fonctionnalités basées sur le contenu de la scène : l'accès, la visualisation et la manipulation par le biais d'interactions multiples. MPEG-4 introduit la notion de codage orienté objet issue du

codage d'image deuxième génération par région. Une scène vidéo se compose dès lors d'une série d'objets audiovisuels codables de manière indépendante. Avec cette norme et pour des débits de 64 Kbit/s à 4 Mbit/s, ce sont tous les services multimédia qui sont concernés : télévision numérique, multimédia mobile, production TV, jeu, vidéo à la demande sur l'Internet (VoD[8]).

Un train binaire MPEG-4 est composé d'objets audiovisuels (AVO) qui peuvent être protégés de manière indépendante. La hiérarchie d'une scène MPEG-4 se décline en plusieurs entités :

- **Video Objet (VO)** : un VO est un objet 2D particulier dans la scène. Rectangulaire ou de forme convexe, il peut indifféremment constituer le fond ou représenter un objet de la scène ;
- **Video Objet Layer (VOL) :** chaque VO peut être codé en multicouches ou en monocouche. Chaque couche de graduabilité est contenue dans un VOL ;
- **Video Objet Plane (VOP) :** un VOP est une instance temporelle d'un VO. C'est une image dans le cas de codage d'objet rectangulaire. Un VOP peut être codé indépendamment des autres, ou prédit par estimation et compensation de mouvement ;
- **Group of Video Objet Plane (GOV) :** le GOV regroupe plusieurs VOP. Les GOV fournissent des points du train binaire où les VOP sont codés indépendamment les uns des autres, et donc offrent des accès aléatoire au train binaire.

L'autre intérêt de la graduabilité MPEG-4 est la différenciation des flux dits de mouvement et de texture.

Le thème de la graduabilité vidéo est ici très succinctement abordé, le lecteur peut cependant trouver un état de l'art plus large sur les deux standards H.263+ et MPEG-4 dans les travaux de thèse de Fabrice Le Léannec [LeLéannec 01].

1.1.3 Robustesse aux pertes de paquets

Observations pour les images fixes

Les deux algorithmes JPEG et SPIHT (§ 1.1.1) ne présentent aucun mécanisme intrinsèque de compensation de pertes de paquets. Le train binaire progressif que fournit les codeurs respectifs est de ce fait sensible à la perte de paquet. L'effacement de la tête du train compromet en effet l'utilisation du reste du flux. Aussi des adaptations aux environnements à pertes de paquets sont réalisées en complément du codage source telles que dans [Goyal 98b] pour JPEG. La forme du train binaire obtenu fait du codage SPIHT une référence dans la transmission progressive d'images fixes ainsi que dans la validation de nombreux mécanismes visant à compenser les effets de pertes de paquets ([Jiang 99]

[8] Video on Demand

[Cosman 00] [Miguel 99] [Mohr 00] [Alatan 00]). Le chapitre 2 est dédiée à l'étude de ces techniques.

En revanche, des outils de résistance à la perte de paquets sont incorporés dans la norme JPEG-2000 [Liang 99]. Ils peuvent être classés selon trois types :

- resynchronisation ;
- rétablissement des données ;
- masquage d'erreur.

Les points de **resynchronisation** ont pour objectif de ne pas corrompre l'ensemble des données d'un train binaire qui suivent une séquence d'effacements. Généralement, la perte se limite entre deux points de synchronisation pour des codes à longueur variable (VLC[9]) (figure 1.3). L'utilisation de codage entropique de type RVLC (*Reversible Variable Length Coded* [Wen 98]) permet cependant d'améliorer, par décodage bidirectionnel, le rétablissement des données non concernées par l'erreur (figure 1.4). L'approche paquet consiste à espacer de manière régulière les marqueurs de synchronisation sur l'image. À débit variable, certaines zones de l'image sont ainsi plus sensibles aux erreurs. Une autre approche consiste à périodiser les marqueurs au niveau du flux binaire. Cette amélioration balise mieux les zones de l'image riches en information (zones de contour). Une dernière approche plus complexe amène à diviser le flux binaire en fonction de la sensibilité des régions aux erreurs et à placer les points de synchronisation en conséquence. De plus, l'organisation hiérarchique du flux binaire JPEG-2000 en sous-bandes, plans de bit et blocs (*Data Partitionning*) cloisonne les erreurs au niveau bloc en réduisant leur propagation. Le **masquage d'erreurs** est du reste des plus rudimentaires puisque, suite à leur localisation, les coefficients erronés sont mis de côté et remplacés par 0.

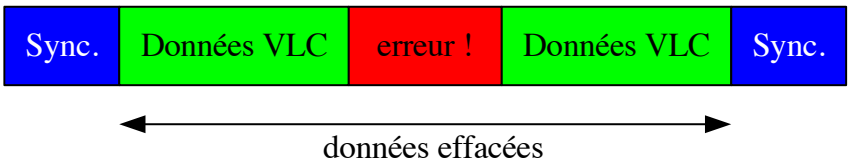


Fig. 1.3: Rétablissement des données par VLC

[9] Variable Length Code

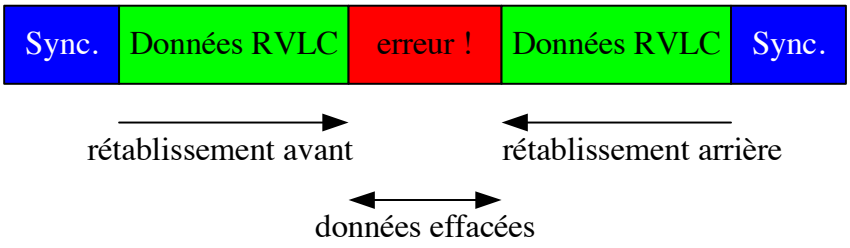


Fig. 1.4: Rétablissement des données par RVLC

L'ensemble de ces outils améliore nettement la robustesse de JPEG-2000 en comparaison de son prédécesseur JPEG [Moccagatta 00]. Cependant, l'ensemble de la syntaxe de JPEG-2000 suppose une transmission graduable de chaque sous-flux avec des paramètres de contrôle d'intégrité propres à chacun, ce que ne peuvent pas toujours fournir les réseaux existants.

Le cas délicat de la vidéo

De manière comparable, MPEG-4 et H.263+ utilisent des marqueurs de synchronisation en cas de train binaire erroné. Lorsque le mode "résistance aux erreurs" est activé dans MPEG-4, un marqueur de resynchronisation (resync-marker ou RM) est inséré dès que le nombre de bits depuis le dernier RM dépasse une quantité fixée. Cet intervalle entre RM est fonction des propriétés du canal et du débit de la source vidéo. La norme définit un paquet vidéo comme l'ensemble des données placées entre deux RM. Le contenu de l'en-tête permet un décodage du paquet indépendamment des autres éventuellement perdus.
Pour les deux normes, une plus redondance accrue est introduite lors de séquences à grand mouvement. Les déplacement des macroblocs sont d'autant mieux résolus temporellement que la fréquence des macroblocs codés en mode INTRA est augmentée. Si l'augmentation ne suffit pas à limiter l'effet des pertes, les macroblocs sont répétés temporellement en dernier recours. Enfin, par l'intermédiaire de messages de retour, le codeur peut être conduit à changer d'image de référence à la suite d'une défaillance au décodage. Cette réaction permet d'éviter la propagation temporelle d'erreur compte-tenu de l'approche prédictive du standard. L'utilisation de cet outil, dans un environnement point-multipoint (multicast) est cependant sujet à conflit entre messages de retour.
Ces deux standards présentent des performances certaines de compression mais de ce fait s'exposent à la propagation temporelle et spatiale des erreurs. La modification des couches application et transport est inéluctable pour assurer une bonne qualité de service à la livraison de vidéo temps réel [LeLéannec 01] [McCanne 96]. En outre, les travaux de Sergio Servetto soulignent l'intérêt de la description multiple abordée dans le chapitre 2 pour ce type d'applications sur des réseaux à contrôle d'intégrité non garantie [Servetto 99a].

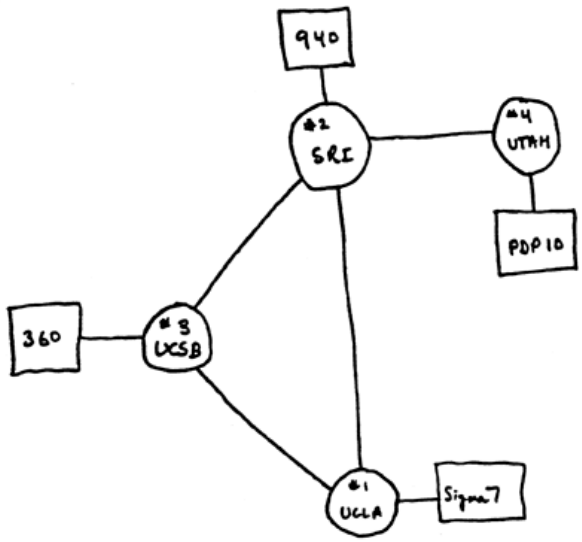


Fig. 1.5: Diagramme des 4-nœuds d'ARPAnet (1969).

1.2 Transmission d'une source graduable sur l'Internet

1.2.1 Vous avez dit réseaux à qualité de service ?

Monde Internet

En 1969, lorsque l'ARPANET est mandaté par le Département de la Défense pour la recherche sur les réseaux, l'Internet se limite à 4 nœuds reliés par des lignes à 50 kbit/s (figure 1.5) :

- l'université de Los Angeles (UCLA) au 1er septembre ;
- l'institut de recherche de Stanford (SRI) au 1er octobre ;
- l'université de Californie de Santa Barbara au 1er novembre ;
- l'université d'Utah au 1er décembre.

Dans cette petite toile, on ne parlait pas de graduabilité mais plutôt de routage aussi rapide que possible (*Hot Potatoes routine doctrine* [Baran 64]) sans que le réseau n'ait connaissance de ce qu'il transportait. La principale préoccupation était la conservation de l'état physique du réseau et sa réaction en cas de la destruction stratégique de nœuds.

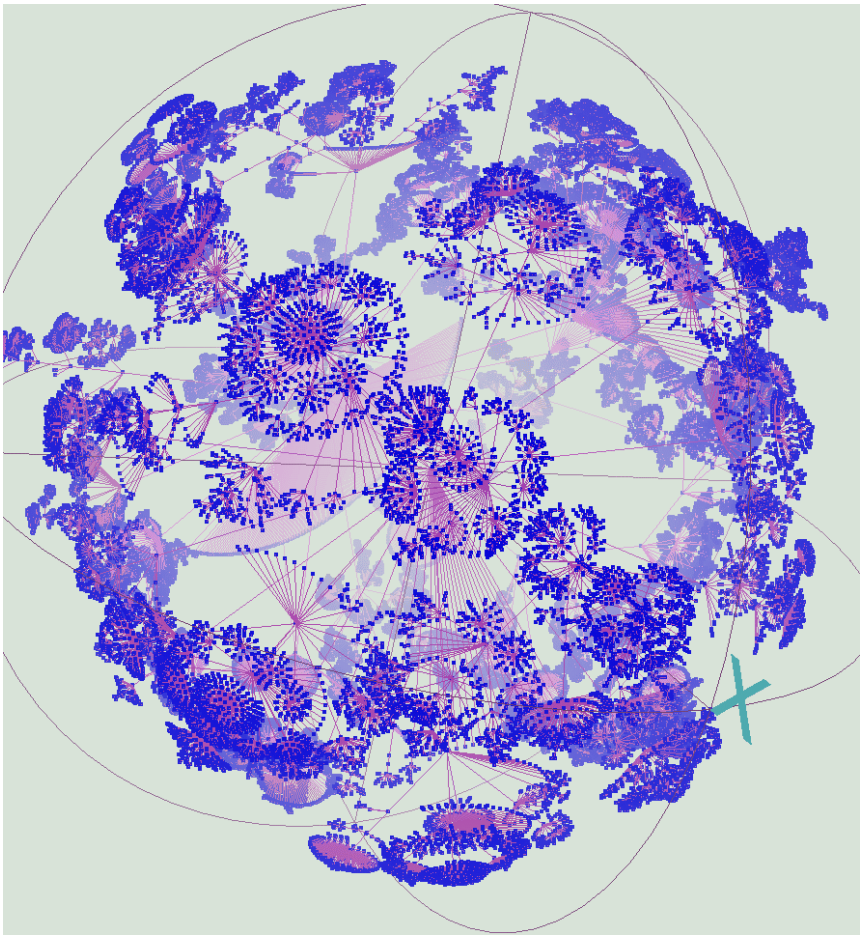


Fig. 1.6: Exploration partielle de l'Internet 2001 à partir du moniteur ihr.skitter.caida.org limitée à 535 102 nœuds.

Aujourd'hui, le problème de la différenciation des flux se complexifie avec les dimensions du réseau [Paxson 97] : 45 millions d'ordinateurs en 1999. Il est prévu 1 milliard d'utilisateurs en 2005. L'ensemble des services entraîne un doublement du trafic toutes les 6 semaines. Le trafic web explose depuis 1992 avec 65000 nouveaux sites web par heure. Devant l'ampleur du phénomène, il semble que la modélisation d'Internet et de manière plus générale d'un WAN[10] piloté par un petit nombre d'invariants et de paramètres relève de l'utopie. C'est la première raison de son succès qui indique la première difficulté. La toile a pour objectif d'unifier les différentes technologies et administrations réseaux. Mais devant l'apparence d'une connectivité uniforme pour les utilisateurs se cache un comportement hétérogène. La deuxième difficulté est la taille de l'Internet. Internet est énorme autant par son nombre de machines que par son hétérogénéité (voir figure 1.6). Les protocoles, écrits souvent pour échelle plus réduite, se comportent dès lors de manière imprévisible. Enfin, les propriétés d'Internet changent en fonction des heures de la journée et des jours

[10] Wide Area Network

de la semaine.

Paramètres de qualité de service

Jusqu'alors, nous avons évoqué le contrôle d'intégrité du réseaux pour désigner sa capacité à protéger un flux lors de sa transmission. Mais l'intégrité n'est qu'un aspect de la qualité de service abondamment décrit dans la littérature ([Pouliquen 00]) et qui est synthétisée dans ce paragraphe. La qualité de service ou **QoS** (Quality of Service) est l'objet d'une quantité de débats importants sur les réseaux, mais dont le traitement reste très confus. Ainsi, le réseau Internet (mode sans connexion, niveau de service "*best effort*") ne permet pas d'offrir une qualité de service constante, ni de donner concrètement des priorités à certains trafics.

La qualité de service est généralement assimilée à la distinction des services ou des flux, et plus encore, à la définition de classes différenciées de service. La garantie d'un service est obtenue en réservant des ressources et en maintenant leur **disponibilité**. Pour mettre en œuvre ces classes ou garantir ces ressources, il convient de définir une ou plusieurs politiques concernants les nœuds du réseau. Finalement, la QoS englobe tous les mécanismes permettant de différencier les types de trafic, ceux ci pouvant être classés et administrés différemment à travers les réseaux.

Parallèlement au développement de réseaux haut débit (ATM, IP sur ATM, 10 Gigabit Ethernet) une nouvelle génération d'applications est apparue avec l'avènement du multimédia. Celles-ci manipulent différents types d'informations (voix, vidéo, graphiques, textes, etc.). Certaines de ces applications se trouvent dans un environnement distribué, où le réseau d'interconnexion impose de nombreuses exigences et contraintes contradictoire liées à la nature des applications, à leurs architectures et aux médias qu'elles manipulent. Les deux types de média manipulés par ces applications sont les médias discrets et continus. Les médias discrets tels que le texte, les images fixes n'ont généralement pas besoin de délais bornés de transmission mais exigent souvent une grande **fiabilité**. Les médias continus tels que la voix, la vidéo sont en revanche, très sensibles aux **délais** d'acheminement et aux variations de ces délais (gigue).

On peut globalement exprimer les besoins des applications multimédias par trois classes de paramètres. Le **débit utile**, qui indique les besoins des applications en quantités de données émises sur le réseau. Ce paramètre peut être variable pour une application et très sporadique à l'usage qui doit donc bien spécifier le débit crête et la sporadicité, en s'engageant sur ses besoins de débit moyen. Le second paramètre est le **délai de transmission** des données qui correspond au temps d'acheminement des données de bout en bout. Selon leur type, les applications exigent des délais et des variations autour de ces délais plus ou moins stricts. Pour les applications temps réel, le délai a une valeur bornée (e.g. 150 ms pour la transmission de la vidéo (G.114)). Ceci permet d'exprimer des exigences en limites absolues ou probabilistes, autorisant le réseau d'agir pour que

ces valeurs ne soient pas dépassées ou le soient avec une probabilité acceptable. Enfin, le dernier paramètre est **la fiabilité de transmission.** Certaines applications tolèrent des pertes d'information non capitale pour le décodage (environ 5% de pertes de paquet pour la voix sur IP). D'autres, comme le transfert de fichier ou de données chiffrées, exigent que toutes les données soient acheminées.

La QoS demandée par une application à un réseau, dépend essentiellement de sa propre QoS vis-à-vis de l'utilisateur. La confidentialité et l'authentification sont des éléments de QoS qu'une application ou qu'un utilisateur peut exiger. Il existe donc deux domaines de QoS dans les environnements où les applications communiquent : la QoS réseau et la QoS applicative.

Caractéristiques réelles du canal

L'intégrité des données transmises sur l'Internet est tributaire principalement de la perte de paquets. On distingue deux situations où peuvent intervenir ces pertes :

- la première consiste à mettre en relation une artère supportant du haut-débit et un utilisateur terminal relié à cette artère par un lien de capacité plus faible (e.g. boucle locale) ;

- la seconde correspond à la diffusion avec des récepteurs de capacités différentes (taille mémoire, processeur, vitesse du bus de transfert).

Henning Schulzrinne, initiateur de RTP[11], prévoyait en 1996 que les pertes sur certaines zones de l'Internet à des instants donnés pourraient atteindre 1 à 10 %. Actuellement, les bulletins météo du réseaux [NOC] donnent parfois des pertes beaucoup plus importantes. Les données ci-dessous sont extraites du Network Operations Center (NOC) de l'université de Los Angeles (UCLA). Les figures 1.7, 1.8, 1.9 et 1.10 illustrent des mesures de perte de paquets à l'aide de la commande "ping" paramètrée par l'envoi de 45 paquets de 210 octets chacun à un débit de 1 paquet par seconde. Les figures 1.7 et 1.8 illustrent une liaison intercontinentale très perturbée entre Los Angeles et Brisbane. Des pics de 50 % environ sont enregistrés pour des temps aller et retour minimum, moyen et maximum respectivement de 430, 451 et 542 ms. Les pertes enregistrées à l'échelle d'un pays sont ici moins importantes (figure 1.9 et 1.10). Les temps de transmission aller et retour sont donnés à 87, 88 et 104 ms.

[11] Real Time Protocol

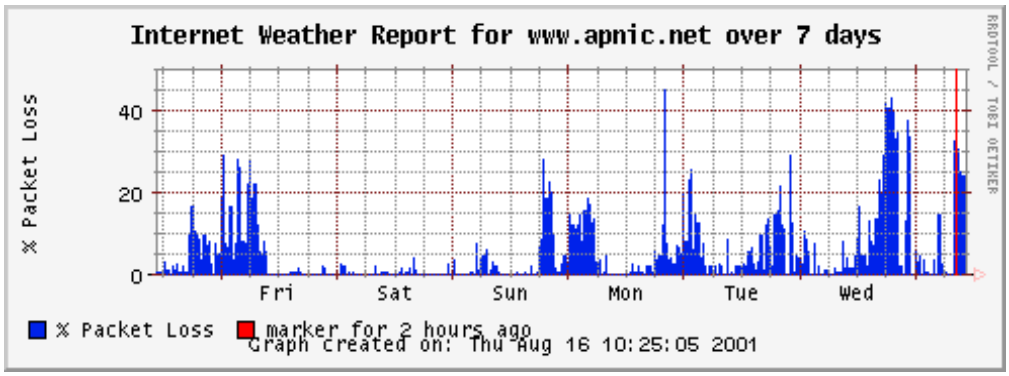


Fig. 1.7: Perte de paquets sur une semaine entre NOC (Los Angeles) et Brisbane (Australie).

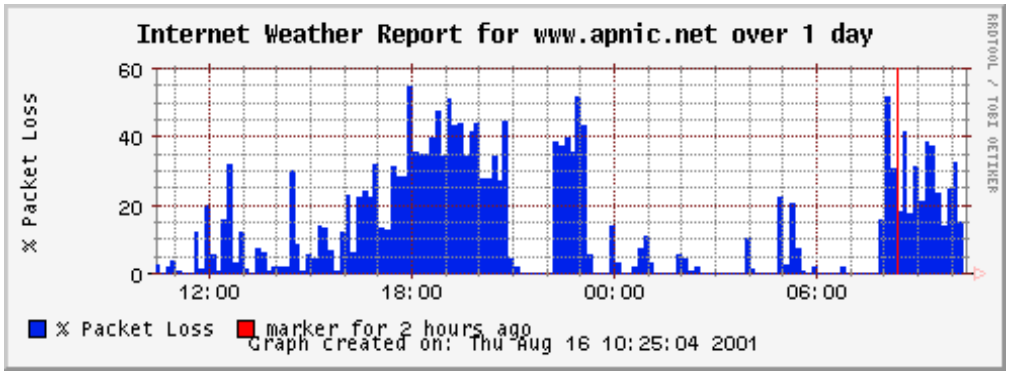


Fig. 1.8: Perte de paquets sur une journée entre NOC (Los Angeles) et Brisbane (Australie).

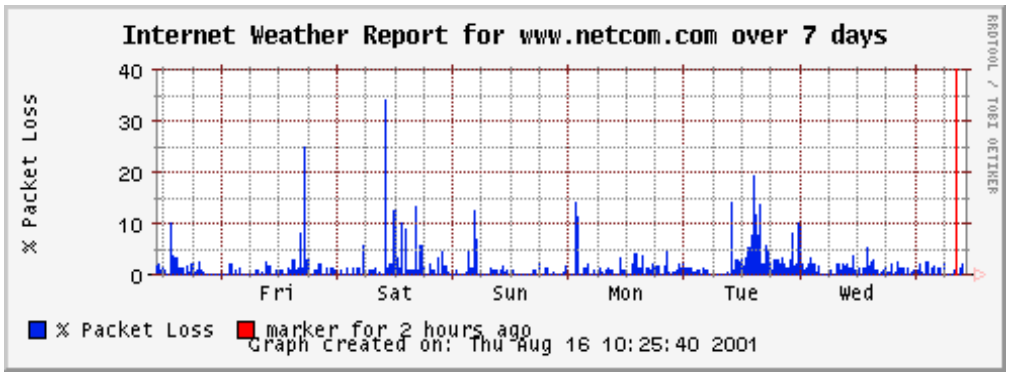


Fig. 1.9: Perte de paquets sur 7 jours entre Los Angeles (CA) et Atlanta (GA).

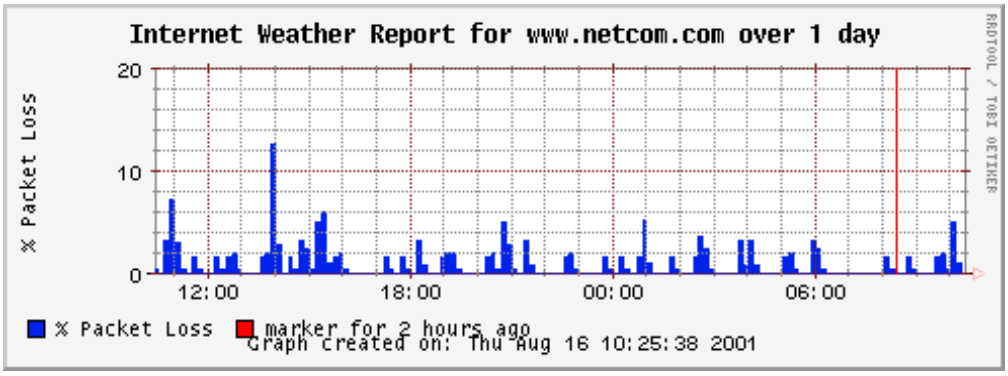


Fig. 1.10: Perte de paquets sur une journée entre Los Angeles (CA) et Atlanta (GA).

Solutions graduables sur IP

Face à ces taux de pertes de paquets, la mesure d'urgence est de sauver les paquets les plus importants. Cette solution implique une gestion de bout en bout des niveaux de priorité qu'affecte un client à chacun de ses paquets.

Les prémices de QoS sur IP sont données avec le champ de 8 bits Type Of Service (TOS) de l'IP version 4. Celui-ci assure un temps de livraison pour deux types de services à savoir un délai normal et un délai court (*Normal Delay* et *Low Delay*) ainsi que deux niveaux de priorité au sein des routeurs. Une extension est ensuite réalisée à quatre et huit niveaux et des éléments de contrôle du réseau ajoutés. Malheureusement, le champ TOS est exploité par un très petit nombre de routeurs IP.
En réservant les ressources nécessaires dans les routeurs qui constituent le chemin, RSVP[12] [Braden 97] permet de maintenir les paramètres de QoS demandés par une application. Son rôle se situe dans le transport des informations de police (pour éviter les réservations intempestives) et de sécurité. Cependant, cette solution réseau développée entre 1994 et 1997 au sein de l'*Integrated Service* de l'IETF[13], avait pour ambition, avec la réservation de ressources de créer un réseau de services intégrés. Chaque source commande les ressources dont elle a besoin, pour un flot déterminé. Mais, dans un réseau à grande échelle, la complexité de l'ensemble a vite remis en cause la viabilité de la solution.

Une alternative IntServ/RSVP est donnée par DiffServ[14] qui, comme son nom l'indique, différencie les flux par classes de service au niveau des équipements d'interconnexions. C'est à chaque nœud d'apporter un traitement différencié, *Per Hop Behaviour*, en fonction de la classe de service du paquet concerné. Il n'est plus nécessaire de maintenir des états dans les routeurs pour chacun des flots. De même , le réseau fait l'économie d'un mécanisme de signalisation tel que RSVP. Chaque nœud du réseau se contente de filtrer les paquets en ne s'intéressant qu'à un seul champ d'en-tête, celui qui porte l'indice

[12] Ressource reSerVation Protocol
[13] Internet Engineering Task Force
[14] Differentiated Service

de sa classe de service. Des files d'attentes distinctes sont alors sélectionnées en fonction de la priorité des paquets. Le protocole reconvertit le champ TOS qui était jusqu'alors inutilisé pour son fonctionnement sur l'IPv4 et utilise le champ DSCP[15] dédié dans IPv6. Son déploiement reste cependant limité et des propositions émergent pour l'incorporation des codes correcteurs par anticipation dans le mécanisme Diffserv/IP [Bennani 01].

1.2.2 Modélisation du canal

Face à la complexité pratique du réseau Internet, plusieurs modèles de canaux émanent de la théorie de l'information répertoriés dans [LeLéannec 01] et [Mohr 00].

Modèles de canaux

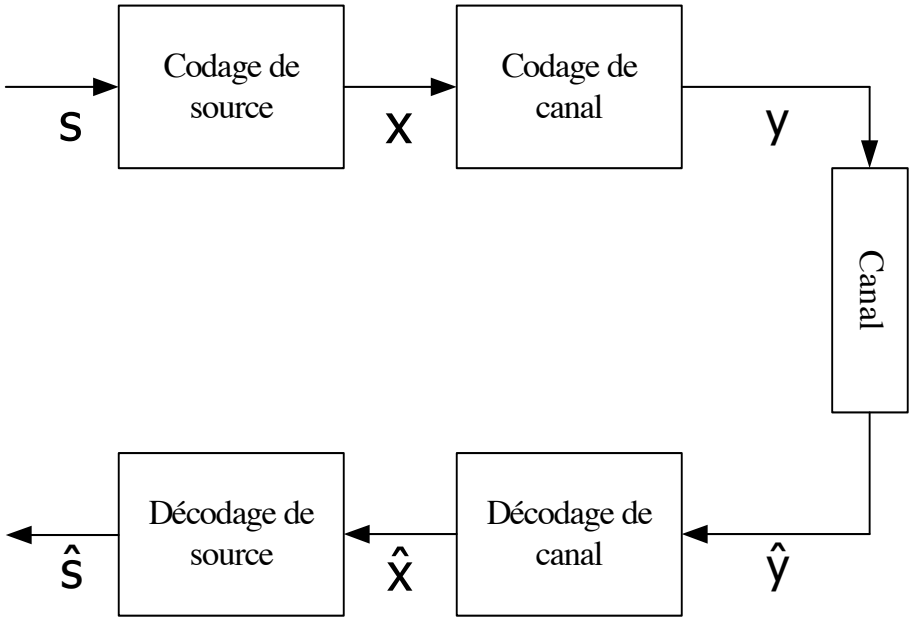


Fig. 1.11: Schéma général de transmission avec séparation du codage de source et du codage de canal.

Notons Y et $\hat{Y}$ les variables aléatoires modélisant un flux binaire à transmettre dont les signaux y et $\hat{y}$ sont des réalisations. Soit N la dimension des vecteurs y et $\hat{y}$. Y prend ses valeurs dans un ensemble $\mathcal{Y} = \{y_1, ...y_n\}$, appelé l'alphabet fini correpondant généralement à une modulation donnée. L'alphabet du récepteur dans lequel $\hat{Y}$ prend ses valeurs, est noté $\widehat{\mathcal{Y}} = \{\hat{y}_1, ...\hat{y}_m\}$.

[15] Differentiated Service Code Point

Définition du canal : *un canal est défini par une famille (W_s) matrice de probabilités conditionnelles de $Y \times \hat{Y} \rightarrow [0,1]$, telles que l'élément (i, j) de la matrice $W_s(y_i,\ \hat{y}_j) = Prob(\hat{Y} = \hat{y}_j | Y = y_{i,s})$ est la probabilité d'obtenir le symbole $\hat{y}_j$ sachant que le symbole y_i est transmis en entrée du canal.*

L'ensemble des probabilités conditionnelles (W_s) modélise l'effet de distorsion du canal. Si $m = n$ les deux alphabets sont identiques, alors l'obtention d'un système de transmission fiable consiste à essayer d'obtenir des valeurs de probabilités $Prob(\hat{Y} = \hat{y}_j | Y = y_i)$ aussi proche que possible de 1. Dans ce cas, le récepteur récupère l'information originale de la source avec une grande probabilité.

Définition du canal sans mémoire : *un canal est sans mémoire si ses lois de transmission sont indépendantes d'une transmission à l'autre.*

Définition du canal à effacement : *un canal à effacement est un canal où l'alphabet de sortie contient en plus de l'alphabet d'entrée un symbole d'effacement $\perp$. Quand un symbole de Y est transmis, il est reçu correctement ou bien il est remplacé par le symbole $\perp$.*

Dans le cas de transmission sur réseau, chaque paquet est un symbole de l'alphabet source. En outre, la définition du canal à effacement ne considère pas les délais de transmission. Dans le cadre d'applications temps-réels, si le délais de réception des paquets est trop important, les paquets sont considérés comme perdus et donc effacés (*Time out*).

Modèle de canal sans mémoire Le modèle de canal le plus simple est le canal à pertes sans mémoire. Dans ce canal, les pertes sont indépendantes et identiquement distribuées. Les paquets arrivent sans erreur avec une probabilité $(1 - p)$, ou bien sont perdus avec une probabilité p. Les événements successifs de réception et de perte sont indépendants. Le processus de pertes peut dans ce cas être modélisé par un processus de Bernoulli.

Canal à mémoire Le canal sans mémoire ne reflète pas fidèlement les processus de pertes observés sur l'Internet. Les pertes en rafales sont mieux modélisées par un canal à mémoire. Le canal à états finis appartient à ce modèle.

Définition du canal à états finis : *un canal à états finis est caractérisé par un espace d'états noté S. À chaque état $s \in S$ est associé un canal sans mémoire $W(.,.,s)$. Conditionnellement à la suite des états, nous considérons des canaux distincts mais dépendants. Cette dépendance est fonction de la manière dont le canal passe d'un état à un autre. La manière dont le canal passe d'un état à un autre détermine le type de dépendance à envisager. Lorsque le futur ne dépend que de l'état courant, le canal est dit markovien*

d'ordre 1, i.e. qu'à chaque état est associée une probabilité sur l'ensemble des états qui détermine la loi conditionnelle de l'état suivant.

Ce modèle est largement utilisé dans la transmission de flux multimédia temps-réels. Notons que considérer ce modèle implique que l'utilisation du canal ne modifie pas son état, ce qui n'est pas toujours réaliste.
La suite des états d'un canal à états finis markovien constitue une chaîne de Markov homogène, définie par sa matrice de transitions $\gamma : S \times S \rightarrow [0, 1]$, et par la définition des canaux associés à chacun des états. Pour chaque $s \in S$, $W(., ., s)$ est appelé la matrice de transfert. En pratique, le processus de pertes est comparable au modèle de Gilbert-Elliott (figure 1.12).

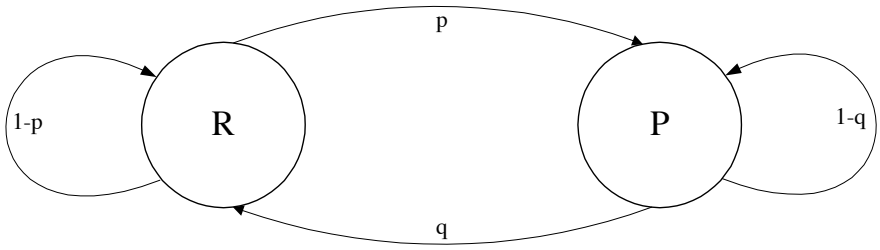


Fig. 1.12: Modèle de Gilbert-Elliott (R : paquet reçu ; P : paquet perdu).

Un modèle exponentiel est aussi retenu plus simplement pour le processus de pertes sur un canal à mémoire. Soit X : le nombre de paquets perdus. Le nombre moyen de paquets perdus parmi N envoyés dans ce modèle est μ pour une densité de probabilité

$$f(X = x) = \frac{1}{\mu} e^{-\frac{x}{\mu}},$$

avec

$$\frac{K}{\mu} \sum_{x=0}^{N} e^{-\frac{x}{\mu}} = 1,$$

où

$$K = \mu \frac{1 - e^{-\frac{1}{\mu}}}{1 - e^{-\frac{N+1}{\mu}}}.$$

1.2.3 Codage conjoint source/canal

Le "théorème de séparation" nécessite tant au niveau de la source que du canal des codes de très grande taille pour réduire suffisamment la probabilité d'erreur. Il n'est ainsi plus valable pour des codes à longueur limitée. De plus, il fut énoncé à l'origine pour une

communication point à point et se révèle inadapté dans l'optimisation d'une communication multipoint comme la diffusion d'un contenu audiovisuel vers un groupe d'utilisateurs. Aussi, la stratégie traditionnelle, consistant à comprimer au maximum la source pour un niveau de distorsion donné, n'améliore pas toujours les performances du système global [Duhamel 97]. Ceci parce qu'elle rend plus sensible le train binaire aux erreurs de transmission : une erreur de peut avoir des conséquences catastrophiques sur la compréhension du signal après décompression. L'optimisation du système nécessite un équilibrage des rendements des codages de canal et de source en prenant en compte les propriétés du canal directement dans le codage de source. Ce faisant, on parlera alors de codage conjoint source-canal de la même manière que l'on parle de fusion entre QoS applicative et QoS réseau. Cette amélioration ne doit pas, cependant, aller à l'encontre de l'évolution des réseaux où différents types des source coexistent. Il est en effet maintenant très rare qu'un réseau donné soit dédié à un seul type de source. Dans ce contexte, le codage conjoint source/canal se doit de préserver l'indépendance de la source vis-à-vis du réseau. Ceci peut être réalisé en partie par des choix judicieux des modèles de canal, et dans une moindre mesure, de la source.

Le schéma de la figure 1.11 reste très général ; le codage de source peut inclure une transformée, une quantification vectorielle et/ou un codeur entropique. Par ailleurs, le modèle du canal peut prendre en compte, outre les caractéristiques de la transmission, tout ou partie des organes d'émission et de réception. On étend donc la définition du canal en englobant dans ce terme les aspects protocolaires inhérents à la transmission. Notre optimisation concerne ainsi principalement la transmission de sources graduables existantes sur le protocole IP.

1.3 Conclusion

Le "théorème de séparation" est sous-jacent tout au long de ce chapitre. Nous sommes passés ainsi par des exemples de codage de source d'image et vidéo et pour le canal. Nous avons abordé les solutions proposées dans un environnement à pertes de paquets tel que l'*Internet Protocol*(IP). De manière générale, la modélisation d'un WAN pilotée par un petit nombre d'invariants et de paramètres reste aujourd'hui un problème très ouvert. On choisit pour la suite de cette thèse le modèle simple du canal à mémoire avec un modèle exponentiel pour le processus de pertes comme référence à un canal IP.
Dans un objectif de transmission, le premier réflexe a été de concevoir des codeurs en réduisant l'impact des erreurs éventuelles sur le canal au décodage. Les codeurs sont alors plus robustes, sans forcément prendre en compte les techniques de codage canal. Cette stratégie peut être améliorée dans un contexte graduable, où les différents éléments transmis n'ont pas la même importance pour restituer la source. Une première approche consiste à maintenir cette hiérarchie au niveau du canal pour que celui-ci applique les priorités né-

cessaires de bout en bout de la transmission (i.e DiffServ). Une seconde approche aboutit à la rupture de la hiérarchie de la source juste avant sa transmission. Le réseau n'est pas de cette façon sollicité dans la gestion des priorités. La source est distribuée en de multiple entités banalisées qui peuvent être effacées arbitrairement. On restitue néanmoins la source de manière graduable au décodage. Ce mécanisme présenté aujourd'hui comme la description multiple de l'information est développé dans notre prochain chapitre.

Chapitre 2

Mécanismes de protection par description multiple de l'information

Sommaire

2.1 Codes correcteurs par anticipation (FEC) 30
2.1.1 Les codes MDS 32
2.1.2 Un autre code proche de la solution optimale 36
2.2 Codage à description multiple 38
2.2.1 Le problème initial 38
2.2.2 Applications 40
2.2.3 Codages à description multiple 40
2.3 Transmission graduable multimédia 44
2.3.1 Protection graduable par description multiple 44
2.3.2 Protection graduable par codage FEC 45
2.3.3 Protocoles de transmission multimédia graduables 46
2.4 Conclusion 48

Le codage de source (la compression) est aujourd'hui parfaitement adapté au stockage sur un seul site d'une version plus ou moins dégradée de la source. Mais lorsqu'une source doit être distribuée, l'application des méthodes classiques de codage de source rencontre quelques difficultés. Le problème est encore plus critique en communication. Certes, un taux de compression élevé réduit fortement le volume à transmettre mais optimiser ce seul paramètre de codage ne préserve en rien la source des aléas du réseaux (pertes de paquets, délais, interruptions...). La perte d'une petite portion du flux source peut en effet rendre inutilisable l'ensemble des informations reçues. Cependant des éléments nouveaux sont apportés de part et d'autre de la frontière historique source et canal pour compenser les effets de perte.

Côté source, une représentation graduable (LC pour *Layered Coding*) sépare en plusieurs couches l'information pertinente de celle de moindre importance. Mais, faisant au mieux, le canal est souvent incapable de gérer la hiérarchie engendrée.
Côté canal, les protocoles de communication utilisés assurent au final, par accusé de réception et par retransmission, une transmission de paquets et ce, quelle que soit la nature du lien (*Transport Control Packet*). Les temps d'aller et retour entre l'émetteur et le récepteur sont néanmoins préjudiciables voire incompatibles à de nombreuses applications temps réel (*Video on demand*, *Voice over IP*...). Une mesure de précaution est alors préferable en incorporant une redondance dans l'information à transmettre. Cette répétition est agencée de manière à créer des paquets utilisables dans n'importe quelle combinaison de réception. En cassant la scalabilité de la source, ce mécanisme est parfaitement applicable à la transmission sur réseaux dépourvus de qualité service. Dès lors qu'un nombre de paquets correctement reçu est suffisant, l'information initiale est reconstruite. En dessous de ce nombre, le niveau de reconstruction est faible voir nul. Chaque paquet de données a ainsi une "valeur" équivalente pour la reconstruction de la source : cette technique, parente du codage canal classique, utilise les codes correcteurs par anticipation (de l'anglais FEC *Forward Error Correcting codes)*. Elle est abordée dans la section 2.1.
Lors d'une transmission par paquets, les niveaux hiérarchiques de la source peuvent être différenciables par le nombre de paquets nécessaires à leurs reconstructions. La réception de chaque paquet permet, dans ce cas, une reconstruction progressive de l'information initiale. Cette approche offre une généralisation du problème de la description multiple présentée dans ce chapitre. Après une formulation du problème, des exemples de codage à description multiple sont développés dans la section 2.2. L'utilisation de ces techniques pour la protection d'une source graduable se concrétise par un système de transmission par codage prioritaire. On précisera en outre l'importance des solutions réseaux existantes. Les deux approches source et canal sont loin d'être antagonistes. Nous tentons dans la conclusion de les réunir pour proposer une méthode de codage conjoint source-canal.

2.1 Codes correcteurs par anticipation (FEC)

Une forme de protection classique pour la transmission sûre de l'information, est l'utilisation des codes correcteurs. Ils sont largement utilisés dans le domaine des transmissions sur canal bruité (satellite, Internet, réseau sans fil avec terminaux mobiles) ou du stockage (compact disque, RAID). Le principal interêt se situe dans l'absence de retransmissions. Une fois envoyé, le volume de données est en effet supposé autonome. L'intégrité est assurée par la redondance introduite à l'envoi. On parlera ainsi de **codes correcteurs par anticipation** ou avant que l'émetteur ne soit sollicité pour retransmettre (FEC pour *Forward Error Correcting codes*). Si le gain de temps est incontestable devant la supression des allers et retours aussi bien des données que des meta-données (acquittements), le gain

en débit introduit pour parvenir à un volume autonome n'est pas évident. Pour illustrer l'avantage des codes FEC, considérons l'exemple suivant. Supposons 10 000 utilisateurs dans une transmission multicast avec un taux moyen de perte de paquets de 10 %. La première solution pour assurer à tous les utilisateurs la bonne réception de tous les paquets est d'envoyer chaque paquet jusqu'à leur acquittement par tous les utilisateurs. Si les pertes sont indépendantes pour les différents utilisateurs, un calcul simple montre qu'en moyenne un paquet doit être transmis approximativement cinq fois. Supposons maintenant que la transmission soit composée de 100 000 paquets et un code FEC est utilisé pour ajouter 25 000 paquets redondants. Si aucun n'utilisateur ne perd plus de 25 000 paquets (*i.e.,* le taux de perte maximum est au plus deux fois le taux de perte moyen (20 %)), alors tous les utilisateurs recevront assez d'informations pour décoder le message original. La bande passante utilisée par la solution FEC est réduite d'un facteur quatre par rapport à la solution avec retransmission. En outre, l'autonomie du volume total de données envoyées rend le protocole globalement beaucoup plus simple.

La redondance incorporée par les codes FEC est systématique. On peut distinguer les paquets de message des paquets de redondance (figure 2.1). Cette forme assure l'utilisation directe des paquets messages pour un utilisateur n'ayant aucune connaissance de la méthode de codage employée. Par ailleurs, dans le cas d'un taux de perte très faible, les paquets de message peuvent être véhiculés vers la couche supérieure sans attendre le flux redondant. Aucune distinction n'est faite cependant au décodage sur la nature du paquet. Qu'ils soient redondant ou message, chaque paquet est équivalent pour la reconstruction. Idéalement, le nombre de paquets nécessaires pour le décodage est égal au nombre de paquets du message original. Les codes MDS présentés dans le paragraphe suivant assurent cette propriété aujourd'hui très convoitée.

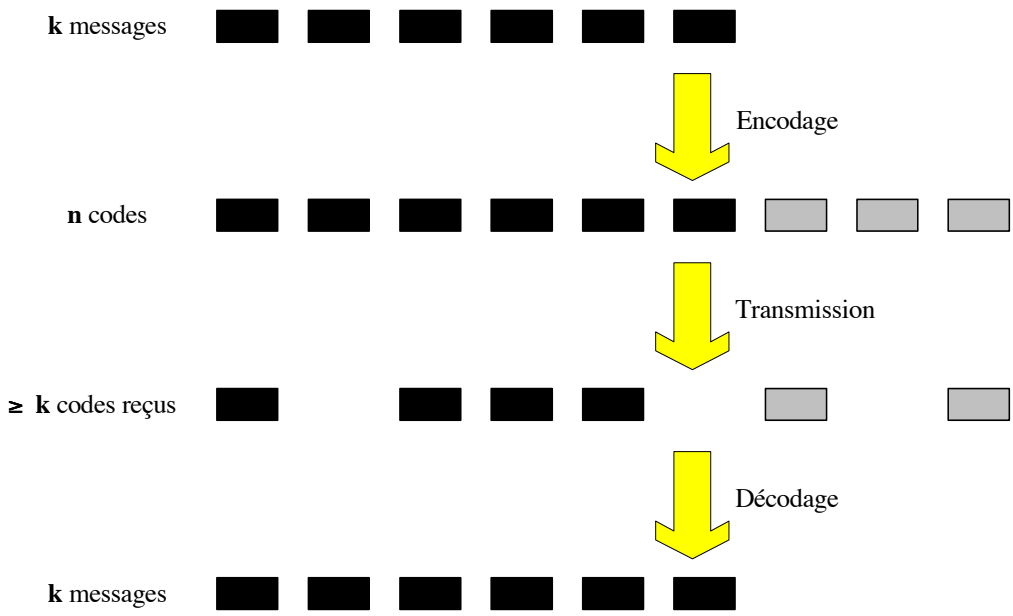


Fig. 2.1: Comment fonctionne un codage FEC ? A partir d'un ensemble de k paquets de message, un ensemble redondant de n paquets est codé ($n \geq k$). La réception d'un ensemble supérieur ou égal à k permet le décodage du message original.

2.1.1 Les codes MDS

Le chapitre 11 de [Williams 77] décrit les codes MDS comme les plus fascinants de toute la théorie des codes. Il est montré que pour tout code linéaire (n, m, d) on a $d \leq n - m + 1$ avec : n, le nombre de symboles codes, m, nombre de symboles message et d, la distance de Hamming. Les codes avec $d = n - m + 1$ sont appelés *maximum distance separable* (MDS). Le nom vient du fait que de tels codes ont une distance maximale entre chaque mot du code et que les symboles du message peuvent être séparés des symboles de contrôle (forme systématique). Pratiquement, la propriété MDS se traduit par le fait que si à partir de m symboles de message d'un corps de Galois GF[2^L], on produit de manière systématique n symboles de codes, n'importe quelle combinaison de m symboles parmi n est suffisant pour décoder le message. La classe la plus importante des codes MDS est celle des codes Reed-Solomon (RS). Les codes MDS sont donc optimaux en termes de nombre de symboles nécessaire au décodage. De plus, sans parler de complexité algorithmique pour l'instant, le décodage est indépendant de l'ordre des m paquets utilisés. Ce codage est applicable aux réseaux de paquets en considérant un paquet comme symbole. On parlera d'un volume d'information fragmenté en m paquets de message et $k = n - m$ paquets de redondance (ou FEC) où au moins m paquets parmi n sont nécessaires pour retrouver le volume original[16].

[16] L'utilisation du code Reed-Solomon [128, 124] dans le protocole ATM au niveau de l'AAL1 illustre cette propriété. 124 cellules sur 128 sont nécessaires pour décoder le volume d'information initial.

Construction

La construction de codes MDS comme proposée dans [Blömer 95] peut s'effectuer par multiplication matricielle d'un mot code (ou paquet de message) par une matrice génératrice.

Soit C une matrice $(n - m \times m)$ sur GF $[2^L]$. La matrice $(I_m|C)$ est génératrice du code MDS systématique avec des paquets de taille L si et seulement si chaque sous-matrice carrée de C est inversible.

Les matrices de Cauchy et de Vandermonde possèdent ces propriétés sur les corps de Galois. On note par [Byers 98] que la mise en œuvre avec une matrice de Cauchy [Luby] est plus rapide pour des grande valeurs de k que la mise en œuvre avec une matrice de Vandermonde [Rizzo 97].

Définition : *Soit F un corps et soit $\{x_1, ...x_m\}, \{y_1, ...y_n\}$ 2 séries d'éléments dans F tel que*
(i) $\forall i \in \{1, ..., m\}$, $\forall j \in \{1, ..., n\} : x_i + y_j \neq 0$.
(ii) $\forall j \in \{1, ..., m\}, i \neq j : x_i \neq x_j$ *et* $\forall i, j \in \{1, ..., n\}, i \neq j : y_i \neq y_j$.

La matrice

$$\left\{ \begin{array}{cccc} \frac{1}{x_1+y_1} & \frac{1}{x_1+y_2} & \cdots & \frac{1}{x_1+y_n} \\ \frac{1}{x_2+y_1} & \frac{1}{x_2+y_2} & \cdots & \frac{1}{x_2+y_n} \\ & & \ddots & \\ \frac{1}{x_{m-1}+y_1} & \frac{1}{x_{m-1}+y_2} & \cdots & \frac{1}{x_{m-1}+y_n} \\ \frac{1}{x_m+y_1} & \frac{1}{x_{m-1}+y_2} & \cdots & \frac{1}{x_m+y_n} \end{array} \right\}$$

est appelée matrice de Cauchy sur F.

Pour chaque entrée de la matrice, on a :

$$\left\{ \begin{array}{ll} x_i = i - 1 & 1 \leq i \leq k \\ & \\ y_i = 2^{L-1} + (j-1) & 1 \leq j \leq m \end{array} \right. \quad 1 \leq i, j \leq 2^{L-1}$$

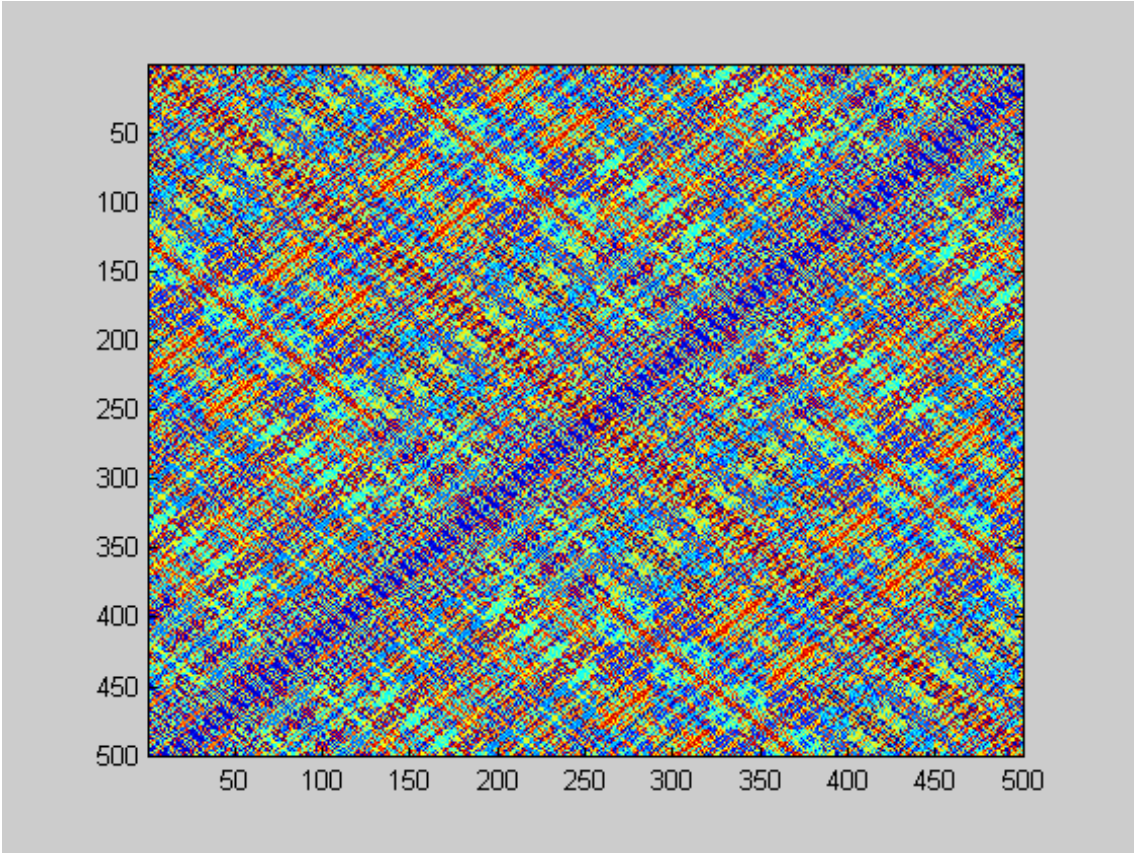


Fig. 2.2: Représentation en niveaux de gris d'une matrice de Cauchy 500×500 sur $GF[2^{10}]$.

Le décodage, à supposer que m paquets soient reçus, s'effectue comme suit.

Soit $I \subseteq \{1, ..., m\}$, la série d'indices de paquets d'information reçus parmi m envoyés. Soit $\bar{I} = \{1, ..., m\} \backslash I$, la série d'indices de paquets d'informations perdus.
Soit $J \subseteq \{m+1, ..., n\}$, la série d'indices de paquets redondants reçus et utilisés.
Par conséquent, $card(J) = card(\bar{I})$ et $card(I) + card(J) = m$.

Le décodage peut se décomposer en différentes étapes :

- contrôler si au moins m paquets sont reçus ;
- collecter tous les paquets d'informations qui sont reçus et remplacer les paquets d'information perdus par les paquets redondants ;
- copier l'information des paquets d'informations dans les parties idoines du message à décoder ;
- calculer D, la sous-matrice carré de C, la matrice de Cauchy originale sur GF[2^L]. Les indices de lignes sont donnés par les indices de paquets redondants utilisés et les indices de colonnes par les indices de paquets d'informations manquants (figure 2.3) ;
- calculer l'inverse D^{-1} de D ;

– mettre à jour les paquets redondants *i.e.* supprimer les contributions des paquets de message effectivement reçus ;

– multiplier la matrice D^{-1} avec les paquets redondants mis à jour pour obtenir les paquets d'information manquants.

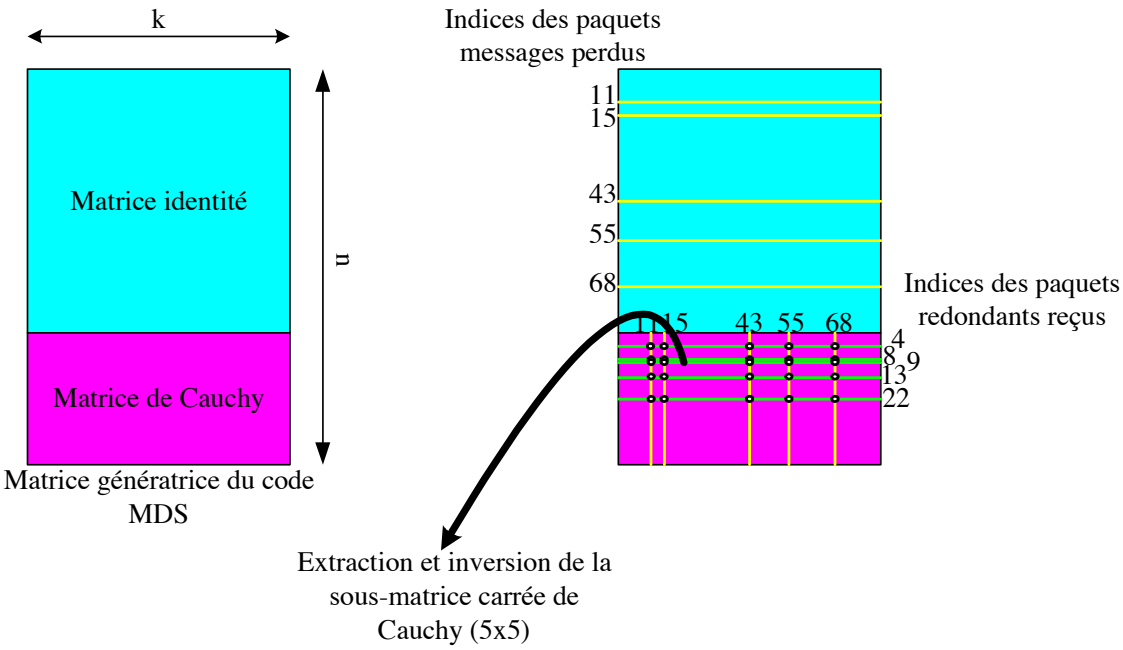


Fig. 2.3: Extraction de la sous-matrice carrée de Cauchy (5×5) pour un scénario de pertes de 5 paquets messages (d'indices : 11, 15, 43, 53, 68) compensées par 5 paquets redondants (d'indices : 4, 8, 9, 13, 22).

Complexité

Le temps du codage est occupé à 99 % par le calcul des paquets redondants. L'affectation d'un identifieur pour chaque paquet et la fragmentation du volume original en paquets de message ne sont donc pas considérées. La complexité s'élève alors à un temps proportionnel à mkL [Byers 98] où m est le nombre de paquets message, k le nombre de paquets redondants et L la taille d'un paquet.

Le temps du décodage est occupé à 99 % par les trois dernières étapes, à savoir l'inversion de la sous-matrice de Cauchy, la mise à jour des paquets redondants et la multiplication matricielle dans $GF[2^L]$. De complexité similaire au codage, la complexité obtenue est de $O(meL)$[Byers 98], où m est le nombre de paquets message, e le nombre de paquets redondants utilisés (le nombre de paquets message perdus) et L la taille d'un paquet. La figure 2.4 extraite de [Blömer 95] donne les temps de décodage en fonction du nombre de paquets redondants reçus (de 1 à 50). Le message à une taille totale de 800 Kb avec une fragmentation de $m = 100$ paquets de message. Chaque opération est effectuée sur

GF[2^{10}].

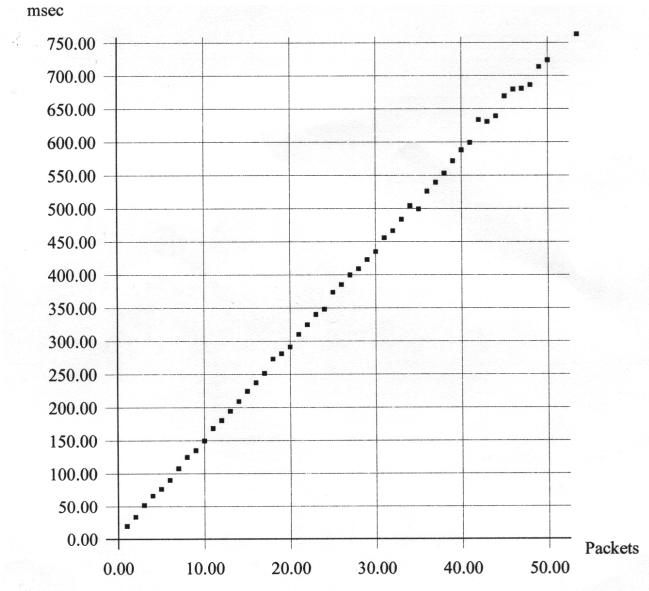


Fig. 2.4: Temps de décodage en fonction du nombre de paquets redondants utilisés

Les codes MDS peuvent être utilisés en pratique lorsque m et k sont relativement petits. Les valeurs utilisées dans [Mohr 00] [Schooler 97] [Hénocq 00] sont en effet entre 8 et 256. En outre, [Blömer 95] spécifie une limite inférieure pour la taille des paquets.

Quel que soit E, un code MDS, qui encode un message de m paquets en un code $E(M)$ de $n \geq m+2$ paquets, la taille L doit satisfaire l'inégalité

$$2^L + 1 \geq \max\{m, n-m\}.$$

2.1.2 Un autre code proche de la solution optimale

La mise en œuvre des codes MDS reste complexe. Un temps de codage plus faible peut être obtenu en nécessitant un nombre de paquets légérement supérieur à m. Ces codes proches de la solution optimale sont appelés $(1+\varepsilon)MDS$ [Alon 96] en référence au nombre de paquets utiles au décodage égal à $(1+\varepsilon)m$. Cette seconde solution allège à la fois le travail d'encodage au niveau du serveur et la reconstitution du volume original chez le destinataire. Les codes Tornado proposés par Luby et al. [Luby 97] sont un bon compromis entre la solution idéale et les temps de codage/décodage.

Construction des codes Tornado

Les paquets redondants sont calculés par combinaisons linéaires des paquets de message ou bien d'autres paquets redondants (figure 2.5). Les matrices utilisées sont très creuses

	Tornado	Reed-Solomon
Coût de décodage	$1+\varepsilon$	1
Temps d'encodage	$n\ln(\frac{1}{\varepsilon})L$	mkL
Temps de décodage	$n\ln(\frac{1}{\varepsilon})L$	meL
Opération de base	XOR	opérations dans corps de Galois

Tab. 2.1: Comparaison entre les complexités des codes Tornando et Reed-Solomon

contrairement aux matrices de Cauchy. Le nombre moyen de variables par équation est faible de manière à opérer par simples substitutions. L'arrivée d'un nombre de paquet légérement supérieur à m entraine une "tornade" de substitutions. En dessous de ce nombre de paquets reçus, le nombre de substitutions possibles est très faible. Les codes Tornado utilisent seulement l'addition modulo 2 et évitent à la fois l'utilisation des opérations de corps et l'inversion de matrices inhérent au décodage Reed-Solomon. Ils sont probabililistes dans la mesure où le message peut être reconstitué par $(1+\varepsilon)m$ paquets avec une grande probabilité.

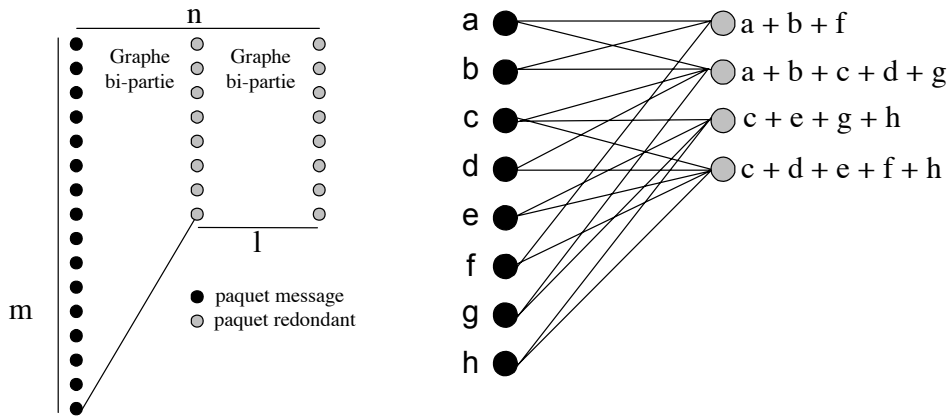


Fig. 2.5: Structure du code Tornado.

Complexité

Pour situer le gain temporel de codage/décodage, le tableau 2.1 résume les complexités du code Tornado et du code Reed-Solomon. Les codes Reed-Solomon ont un temps d'encodage proportionnel à mkP et un temps de décodage proportionnel à meL. Les codes Tornado ont une complexité de l'ordre de $O((m+k)\ln(\frac{1}{\varepsilon})L)$ avec un coût de décodage de $1+\varepsilon$. [Byers 98] donne un exemple significatif d'un code Tornado avec $1+\varepsilon \approx 1,054$ en moyenne. Le coût maximum de décodage s'éléve à $1,10$ pour un écart type de $0,0073$

Le tableau 2.2 de [Byers 98] résume les tests de performance au décodage pour ces deux codes. Pour décoder les codes Reed-Solomon, il est supposé que $m/2$ paquets de

TAILLE	code RS	code Tornado
250 KB	2.06 sec.	0.18 sec.
500 KB	8.4 sec.	0.24 sec.
1 MB	40.5 sec.	0.31 sec.
2 MB	199 sec.	0.44 sec.
4 MB	800 sec.	0.74 sec.
8 MB	3166 sec.	1.28 sec.
16 MB	13629 sec.	2.27 sec.

Tab. 2.2: Temps de décodage des codes Tornado et Reed-Solomon

l'information originale et $m/2$ paquets redondants sont utilisés. Les codes Tornado ont un coût de décodage de 1.054, aussi il est supposé que $1,054 \times m/2$ paquets de message et $1,054 \times m/2$ paquets redondants sont utilisés.

2.2 Codage à description multiple

Le codage à description multiple a été inventé dans le laboratoire Bell. Il part d'une motivation pratique telle que la transmission de la parole sur le réseau téléphonique, devient un problème théorique majeur pour enfin revenir vers une application en sciences de l'ingénieur. La plupart des travaux originaux n'ont malheureusement pas été archivés. Vivek Goyal dans [Goyal 01b] nous livre quelques pièces manquantes. Dans les années 70, le mécanisme envisagé pour prévenir les interruptions téléphoniques est de transmettre un simple appel en le divisant sur deux liens distincts. Dans des conditions normales, les deux canaux peuvent être combinés pour améliorer la qualité de la conversation. Dans le cas contraire, la qualité est certes diminuée mais reste acceptable. S. E. Miller [Miller 78] propose d'améliorer une technique très simple. Par sa forte décroissance spectrale, le signal de parole initialement échantillonné à la fréquence de Nyquist peut être décimé par deux sans trop d'effets d'aliasing. Les échantillons pairs sont alors transmis sur un canal et on transmet sur un autre les échantillons impairs. En 1978 et 1979, l'idée de la séparation de canaux gagne deux groupes au Bell Labs : le codage de la parole et la théorie de l'information. A. Gersho est situé dans ces deux camps. Il partage son problème en septembre 1979 avec S. Jayant, L. Ozarow, H. S. Witsenhausen, J. K. Wolf et A. D. Wyner au cours d'un Workshop sur la théorie de l'information, point de départ de l'étude théorique de la description multiple.

2.2.1 Le problème initial

Les premières publications concernent le problème de deux sources vers trois destinations [Wolf 80], [Ozarow 80]. En voici une description actualisée [Goyal 01b]. La description multiple avec deux canaux et trois récepteurs est schématisée sur la figure 2.6.

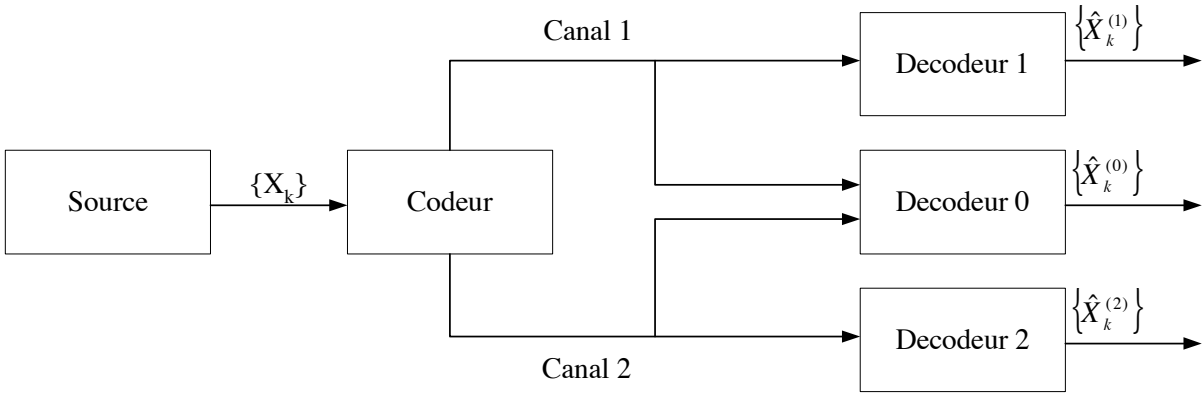


Fig. 2.6: Partage de l'information sur deux sources avec trois destinations.

Une séquence de symboles source $\{X_j\}_{j=1}^{I}$ doit être envoyée à trois récepteurs sur deux canaux bruités. Le décodeur 0 appelé le décodeur central reçoit simultanément les informations des deux canaux alors que les deux décodeurs latéraux recoivent seulement l'information de leur canal respectif. Le débit de transmission sur le canal i est noté R_i, $i = 1, 2$, en bits par échantillon source. La séquence reconstruite produite par le décodeur i est noté $\{\hat{X}_j^{(i)}\}_{j=1}^{I}$, et les distorsions mesurées à ces niveaux de reconstruction sont notées D_i, $i = 0, 1, 2$. La difficulté se situe dans la complémentarité des descriptions. Si on obtient une bonne description à un débit de R_1 à envoyer sur le canal 1 et une autre description à un débit R_2 à envoyer sur le canal 2, il n'y a aucune raison *a priori* pour que la combinaison des deux descriptions améliore la reconstruction pour un débit total de $R_1 + R_2$. De la même façon, il est difficile de partager une source compressée à un débit $R_1 + R_2$ en deux descriptions indépendantes et décodables. Construire des descriptions individuellement acceptables mais pas trop similaires est le compromis fondamental du codage à description multiple. Le problème consiste donc à obtenir, pour une source et une mesure de distorsion données, le meilleur quintuple $(R_1, R_2, D_0, D_1, D_2)$. Michaël Orchard [Orchard 97] précise l'objectif en essayant de minimiser de façon conjointe les deux mesures de distorsion :

- D_0 : la distorsion de la reconstruction à partir des deux canaux ;
- D_1, D_2 : distorsion moyenne de la reconstruction à partir d'un seul canal étant donné une équiprobabilité de pertes d'un des deux canaux.

On parle de description multiple équilibrée si les deux descriptions latérales ont même débit et introduisent la même distorsion (*Balanced Multiple Description Coding*). Les limites théoriques en termes de débit distorsion ont été exprimée par El-Gamal et Cover [El-Gamal 82] ainsi que par Ozarow [Ozarow 80]. Dans [El-Gamal 82], les auteurs montrent les débits réalisables $[R_1, R_2]$ pour un vecteur de distorsion donnée $[D_0, D_1, D_2]$, pour une source arbitraire sans mémoire et pour une mesure de distorsion donnée.

2.2.2 Applications

L'application sur réseau de paquets de la description multiple part de plusieurs analogies. Chaque paquet est considéré comme un canal dans la mesure où le paquet est supposé intact si il est bien reçu. De même, le problème initial spécifie le transport de chaque description sur un canal distinct. Aussi, chaque description est transportée par un paquet entre un émetteur et un ou plusieurs récepteurs. La description multiple permet de ce fait de compenser les pertes de paquets localisées au niveau des points de congestion du réseau (*e.g.* nœuds entre deux liens de capacités différentes) et de prendre en compte les capacités des différents récepteurs. L'approche classique du codage conjoint source-canal (hiérarchisation dans la supression des flux) induit un surcoût de travail à la fois pour le réseau (distinction de chaque paquet dans un nœud) et pour l'utilisateur (hiérarchisation et envoi dans différents sous flux). Dans le codage par description multiple, on supprime le travail du réseau en faisant toujours l'hypothèse que les paquets sont d'égale importance. Le réseau fait ainsi de son mieux pour acheminer les paquets comme le fait au quotidien l'*Internet Protocol* v4.
La description multiple supporte toutes les données multimédias. La transmission d'image sur réseaux hétérogènes est la première cible [Goyal 98b] [Ng 97b] avec parfois une granularité dans la protection de la source excellente [Servetto 00]. Cette granularité permet à des terminaux d'avoir une visualisation rapide de l'image en fonction de leurs ressources en environnement *multicast*. De la même manière, la transmission audio est réalisée dans [Arean 00]. Enfin, par une légère modification du protocole TCP (élimination des retransmissions, acquittements partiels), la description multiple est utilisée pour fournir un contenu vidéo à des utilisateurs de l'Internet public [Servetto 99b]. Sergio Servetto et al. préconisent dans ce dernier papier l'utilisation du filtrage en sous-bandes le long de l'axe temporel (sous-bandes 3D) devant la compensation de mouvement. Cette dernière méthode offre moins de possibilités dans la granularité des débits, des tailles d'images et le débit de trames malgré une meilleure efficacité en compression.

Un peu plus oubliée, l'application dans le stockage distribué de données semble prometteuse avec la description multiple [Guédon 01a]. Elle peut en effet apporter à la fois sécurité et graduabilité aux utilisateurs et aux fournisseurs de contenus. Nous développerons cette application au chapitre VII.

2.2.3 Codages à description multiple

Quantification scalaire à description multiple

Considérons un premier exemple de réalisation de codage à description multiple par une loi de quantification scalaire [Goyal 01b]. On souhaite transmettre un nombre réel dans l'intervalle [-1,1] avec 4 bits. Le choix normal et optimal si la source est uniformément distribuée sur l'intervalle, est d'utiliser la loi uniforme de quantification présentée sur la

figure 2.7(a). Mais cette loi n'est pas appropriée à la séparation vers 2 canaux. Aucune méthode ne permet en effet de scinder les quantificateurs pour la transmission sur 2 canaux en ayant une bonne estimation par un seul canal. Par exemple, si les deux bits de poids fort sont envoyés sur le canal 1 et les deux bits de poids faible sont envoyés sur le canal 2, l'erreur moyenne de reconstruction du canal 2 sera très importante. Une manière simple de distinguer les informations des deux canaux est montrée à la figure 2.7(b). Deux lois différentes sont utilisées pour chaque canal, qui par combinaison, offrent une résolution de 4 bits au décodeur central. De manière générale, avec des quantifications de B-bit, le décodeur central a une résolution de (B+1)-bit. Le taux de redondance élevé (2B au total pour une résolution de (B+1)-bit) motive la recherche d'une autre solution.

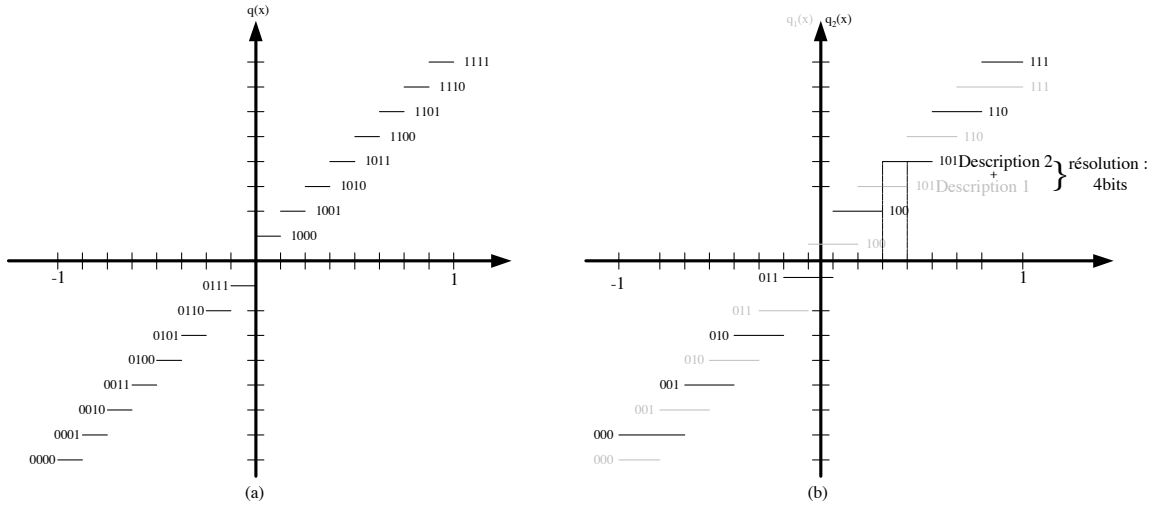


Fig. 2.7: (a) une quantification uniforme de 4 bits. Il n'y a aucune manière pour sectionner les quantificateurs en deux paires de bits pour obtenir une solution acceptable dans les décodeurs latéraux. (b) deux quantifications à 3 bits qui se complémentent pour donner une résolution de 4 bits au décodeur central.

Reudink [Reudink 80] a inventé plusieurs autres techniques avec une plus faible redondance. Jiang et Ortega [Jiang 99] améliore la méthode de Miller [Miller 78] en proposant d'utiliser plusieurs lois de quantification en fonction de la parité des échantillons. Indépendemment, Vaishampayan dans [Vaishampayan 93] a developpé une méthode simple de quantification scalaire à description multiple. Cette méthode se décompose en deux parties principales : un quantificateur scalaire (qui représente des variables aléatoires continues par un ensemble fini d'échantillons), et une allocation d'index (qui sépare l'information de chaque échantillon dans deux descriptions complémentaires d'un même échantillon). L'allocation d'index est une injection $\Psi : \mathsf{N} \longrightarrow \mathsf{N} \times \mathsf{N}$ (N est l'ensemble des entiers naturels). Si la quantification scalaire représente la source par un nombre fini de points (soit I), l'injection Ψ peut être visualisée comme une matrice de taille $(I \times I)$, où seulement I positions sont utilisées. Les quantificateurs sont placés sur la matrice et les index cor-

respondant aux numéros de lignes et de colonnes sont envoyés sur chaque canal. Si les deux descriptions sont disponibles, l'information originale est retrouvée avec une distorsion correspondant seulement au bruit de quantification. Sinon, l'information originale est parmi les élements de la ligne ou de la colonne reçue. Cette idée est illustrée sur la figure 2.8 avec des quantificateurs très concentrés sur la diagonale de la matrice d'allocation. Les distorsions latérales seront d'autant plus élevées que ces quantificateurs seront étalées sur la diagonale. Un plus grand nombre de quantificateurs est disposé sur chaque ligne et chaque colonne sur la figure 2.9. La construction d'une large classe d'allocation d'index est proposée dans [Vaishampayan 93]. Cette classe est utilisée dans [Servetto 00] dans la protection scalable des sous-bandes d'une décomposition ondelette.

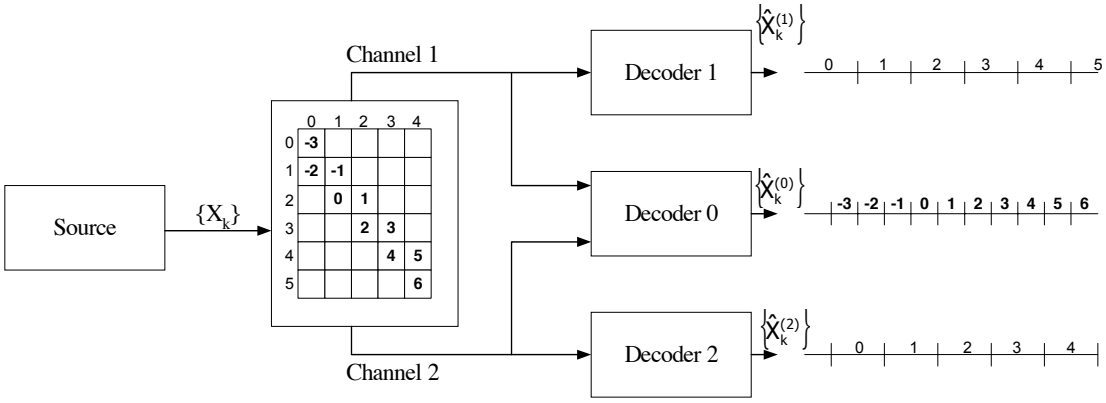


Fig. 2.8: Quantification scalaire à description multiple. Un exemple d'allocation d'index pour les coefficients quantifiés.

-9	-8					
-7	-6	-4				
	-5	-3	-2			
		-1	0	2		
			1	3	4	
				5	6	8
					7	9

Fig. 2.9: Autre exemple d'allocation d'index pour les coefficients quantifiés.

Pour la création de plus de deux descriptions, T. Y. Berger-Wolf et E. M. Reingold généralisent la construction d'allocation d'index proposée ci-dessus, à un tenseur à N-dimension [Wolf 99]. Pour le codage de source à mémoire, Batllo and Vaishampayan construisent des codes à transformation à description multiple basés sur les quantifications à description multiple [Batllo 97]. Leur principal résultat est que, pour une source gaussienne stationnaire, la transformation optimale et l'allocation optimale de bit est identique à un codeur

fournissant une seule description.

Transformations corrélantes

Un autre champ d'investigation dans la construction des descriptions a été initié par Wang et al. dans le but de rendre robuste un codeur JPEG aux erreurs de canal [Wang 97]. L'idée de base est de supprimer toute corrélation contenue dans l'information pour ensuite introduire de nouvelles relations entre les paires de variables aléatoires non corrélées obtenues. Chaque variable est envoyée au final sur un canal séparé. Si une variable est effacée, elle peut être statistiquement estimée par celle reçue. Orchard et al. [Orchard 97] [Wang 98] construisent ces transformations à descriptions multiples en deux étapes successives :

- la première consiste à supprimer toute la corrélation du message original. Cela est fait de façon optimale par une transformation de Karhunen-Loeve (KL) ;
- la seconde étape groupe par deux les éléments issus de la transformation KL (par ex. X_1 et X_2) et applique une transformation 2×2 (une matrice de rotation) qui recorrèle ces deux coefficients avant d'envoyer les deux nouveaux éléments (Y_1 et Y_2) sur chacun des canaux.

Supposons X_1 et X_2 deux variables aléatoires gaussiennes indépendantes, centrées de variances σ_1^2 et σ_2^2, $\sigma_1^2 \neq \sigma_2^2$. La transformation proposée dans [Wang 97] utilise :

$$\begin{aligned} Y_1 &= \frac{1}{\sqrt{2}}(X_1 + X_2) \\ Y_2 &= \frac{1}{\sqrt{2}}(X_1 - X_2) \end{aligned}$$

comme descriptions du vecteur (X_1, X_2). (X_1, X_2) peut être reconstruit à partir des deux descriptions. Les descriptions Y_1 et Y_2 ont pour coefficient de corrélation $(\sigma_1^2+\sigma_2^2)^{-1}(\sigma_1^2 - \sigma_2^2)$. Ainsi, lorsqu'une description est perdue, la description reçue autorise une estimation de la manquante. La transformation KL entraine un sur-débit non négligeable par la transmission des vecteurs de base aussi [Goyal 98a] préconise l'utilisation de la DCT et [Ng 97a] l'analyse en ondelettes. De plus, Goyal et Kovačević généralisent cette construction en utilisant des vecteurs de dimension N au lieu d'une paire de coefficients et ce avec une classe plus large de transformations corrélantes [Goyal 98a] [Goyal 01a]. Au vu des résultats de [Goyal 01a], les transformations corrélantes sont plus optimales que celles de [Vaishampayan 93] pour un taux de redondance faible. Pour des canaux très bruités, la quantification scalaire à description multiple est plus performante. Le choix de la méthode nécessite donc une connaissance de l'état du canal (voies de retour).

2.3 Transmission graduable multimédia

Par transmission scalable multimédia, on sous-entend une protection inégale de la source. Chaque flux source est protégé de manière distincte en fonction de son niveau de priorité. Plusieurs approches, globalement conjointes source-canal, sont aujourd'hui proposées. Si on se refère à la source dans ce qui est présenté ici, celle-ci a un rôle dont l'importance diminue au fil des paragraphes. En effet, elle est directement impliquée lorsque l'on ajoute de la redondance en fonction de sa hiérarchie (codage FEC et description multiple). En outre, son rôle devient négligeable dès lors que des mécanismes réseaux sont entrepris La redondance multi-niveaux peut être effectuée par les méthodes de description multiple introduites à la section 2.2 ou par des codes FEC de pouvoir de correction graduable. En parallèle, des protocoles de transmission multimédia graduable émergent dénotant une approche réseau du problème de gestion hiérarchique de la source.

2.3.1 Protection graduable par description multiple

Comme précisé dans la sous-section 2.2.2, la description multiple est largement appliquée au domaine de l'image. Cependant seulement quelques codages offrent une généralisation du problème pour un nombre de canaux supérieur à deux (*Generalized Multiple Description Coding - GMDC*). L'approche classique est une profonde modification du codage de source de manière à produire non pas un seul flux mais un ensemble de sous-flux multiples indépendants et d'égale importance. De cette façon, la qualité des images reconstruites dépend seulement du nombre de paquets reçus. Le travail du canal se trouve alors fortement réduit car la structure des sous-flux ne révèle aucune hiérarchie de la source. Cette idée est une généralisation du problème de raffinement successif de l'information source où une image peut être divisée en deux descriptions p_1, p_2 [Equitz 91]. La différence dans ce cas, est que la description p_2 est le strict raffinement de la description p_1 en supposant la réception de p_1 ; si p_1 est perdu, à l'inverse, p_2 ne peut pas être utilisé même si celle-ci arrive intact.
Le point commun des méthodes proposées dans la transmission graduable d'images par description multiple est l'utilisation préalable des transformations classiques de codages de source *i.e* transformation en cosinus discret [Goyal 98b] ou ondelettes [Servetto 00]. Goyal et al. proposent une extension à N canaux par une mise en cascade des transformations corrélantes. Celles-ci sont optimisées au cours d'une phase d'apprentissage. Dépendant fortement du contenu de l'image, la description multiple est appliquée à chaque sous-bande d'une image [Servetto 00] [Srivinasan 98]. Il est à noter que cette approche n'est pas une généralisation de la description multiple même si plusieurs canaux sont sollicités. En effet, une description multiple à 2 canaux est appliquée à chaque sous-bande de l'image et non pas sur l'ensemble de son spectre. Dans [Servetto 00], le nombre de diagonales de la matrice d'allocation de [Vaishampayan 93] (exemples sur les figures 2.8 et 2.9) augmente

lorsque l'importance de la sous-bande diminue. La détermination d'un coefficient ondelette basse fréquence sera d'autant plus facilité dans le cas de perte d'une description que le nombre de quantificateurs sur une ligne ou une colonne sera faible. En outre, des codes correcteurs de pouvoir de correction graduable sont appliqués à chaque description en fonction de la sous-bande considérée. La basse fréquence est dotée d'un code à répétition (code MDS trivial) pour sa protection alors que la plus haute fréquence est dépourvue de toute redondance. Avec une dernière étape de codage arithmétique sur chaque mot code, un rapport signal à bruit crête (PSNR) similaire à celui de [Srivinasan 98] et de [Wang 97] est obtenu en utilisant 50% à 60% du débit imposé par ces codeurs.

2.3.2 Protection graduable par codage FEC

La redondance multi-niveaux est ici assurée par des codes correcteurs standards comme présentés au paragraphe2.1. Ces codes auront un pouvoir de correction inégal en fonction de la couche qu'ils protègent (*Unequal Error Protection - UEP*). Les méta-informations qui transitent du codeur source au codeur canal sont limitées généralement aux incréments de qualité qu'apporte chaque couche. Ces méthodes de protection supposent donc une source graduable compressée et utilisent de ce fait les codages de source standards (JPEG, MPEG, SPHIT...) sans opérer de profondes modifications. Albanese et al. proposent un système complet qui avant la transmission gére les priorités de chaque sous-flux source [Albanese 96] (*Priority Encoding Transmission - PET*). Dans la mise en œuvre [Leicher 94], l'utilisateur spécifie le niveau de fragmentation du message source en m paquets, la longueur l de chaque paquet, et une fonction de priorité ρ, et le système calcule le nombre de paquets n ainsi que les procédures de codage et décodage. Ils préconisent l'emploi des codes MDS avec m paquets parmi n suffisants pour la reconstruction. Cette reconstruction est assurée de manière fiable ou avec une haute probabilité selon que le système proposé est deterministe ou probabiliste. De nombreuses réalisations émanent de la publication de ce travail en 1996. Un premier calcul dynamique des niveaux de protections est fournit par [Davis 96] en minimisant la distorsion totale. Le schéma de compression manque néanmoins de performances dans les débits disponibles. Mohr et al. choisissent la protection inégale du codage source SPIHT [Mohr 00] en employant des codes RS dans un environnement de transmissions ATM. Dans l'exemple proposé, la granularité de la description d'une image 512×512 est composée de 47 sous-flux en relation avec la taille d'une cellule ATM avec 23 codes RS distincts. Le gain du PSNR est évident par rapport à une protection globale identique. De la même manière mais pour une source vidéo (H263+), [Hénocq 00] applique les codes RS avec $n = 32$ et $k \in [16, 32]$. En outre, le système est construit pour répondre aux contraintes temps-réel inhérentes à la distribution vidéo. Pour ce même média, [Byers 98] fait un parallèle entre l'indiscernabilité des paquets et des fragments d'eau que peut délivré une fontaine. Quel que soit l'ordre et quel que soit le fragment, une fois que le volume est rempli vous pouvez consommer. Afin de réduire la

complexité, les auteurs utilisent les codes Tornado pour cette "fontaine numérique". Ces deux dernières solutions abordent l'écriture d'un protocole de distribution vidéo sans toutefois confier la gestion de la hiérarchie au réseau. Les fonctionnalités proposées peuvent cependant ne pas être limitée à une couche inférieure de l'applicatif mais être déployé dans tout les nœud du réseau qui constitue le chemin (*Per Hop Behavior*).

2.3.3 Protocoles de transmission multimédia graduables

L'approche de la communauté réseau confie naturellement la protection graduable de la source au réseau lui-même. L'objectif est de résoudre le problème d'hétérogénéité rencontré dans les environnements multicast (figure 2.10). Le réseau ne transmet seulement qu'un nombre de couches que chaque lien physique et chaque terminal peut supporter (l'hétérogénéité ne qualifie pas seulement le réseau mais aussi les terminaux). Par exemple, le particulier ne recevra chez lui que la couche de base au travers d'une ligne ISDN (*Integrated Switch Data Network*). L'utilisateur dans l'Internet MBone [Deering 91] obtiendra une couche supplémentaire alors que l'utilisateur local via un réseau Ethernet 10 Mb/s recevra la totalité du codage de source.

L'objectif se résume dans la capacité du réseau à régler sélectivement le niveau de la requête (dénommée souscription) aux couches constituants la source pour chaque utilisateur. Deering marque une avancée conséquente en proposant un schéma pratique de transmission graduable [Deering 93]. Cependant cette solution manque d'algorithmes spécifiques tant au niveau de l'application que du canal. L'idée que les propriétés du signal doivent orienter la construction du protocole de transport est pour la première fois défini dans l'architecture ALF (*Application Level Framing*) de Clark et Tennenhouse [Clark 90]. En revanche, l'idée que les codeurs impliqués dans l'application nécessitent des propriétés de codage conjoint source canal a été proposé en premier par Garrett et Vetterli [Garrett 93]. McCanne dans sa thèse développe que les deux idées sont en fait des concepts parallèles en proposant le protocole RLM (*Receiver-driven Layered Multicast*) pour la distribution vidéo sur le MBone [McCanne 96]. Dans le protocole RLM, les récepteurs spécifient au réseau l'abandon ou la souscription d'une couche et ce en fonction de l'état du lien avec la source. Cette dernière n'est concernée que dans la délivrance de son signal avec un degré de qualité maximal. Si le niveau de souscription est trop haut, la qualité subjective se trouve altérée par l'effacement de paquets vidéo au niveau d'une congestion. Le réseau réagit alors par l'effacement de la dernière couche. Si le niveau est trop bas, la position du récepteur par rapport au point de congestion est plus difficile à évaluer. Aussi à la manière du *slow-start* pour une source TCP [Jacobson 88], le réseau au niveau des points d'interconnexion aura tendance à transmettre le signal en augmentant progressivement le nombre de couches à la demande du récepteur. La figure 2.11 résume le fonctionnement du protocole. Les ajouts de couches sont rythmés par une temporisation (*join-timer*) qui augmente ou diminue leur fréquence en fonction de l'observation d'une congestion. Une phase

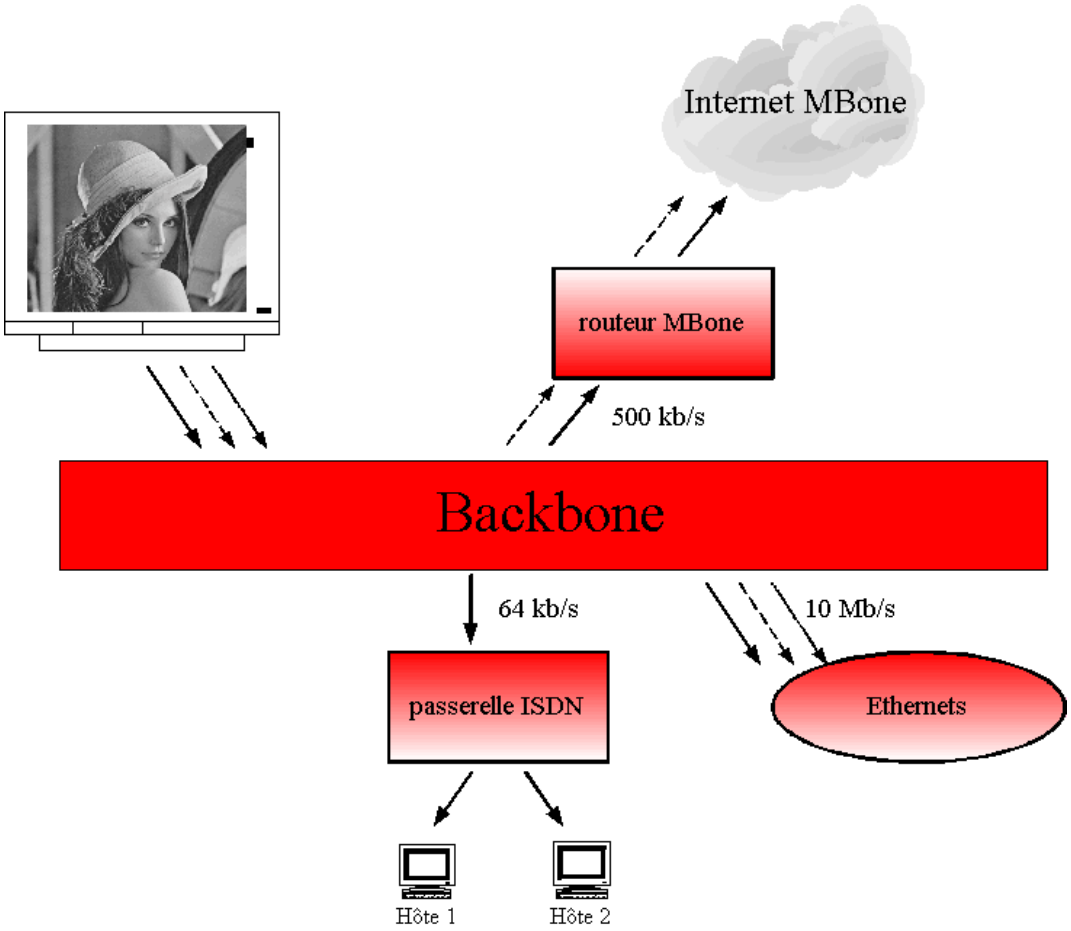


Fig. 2.10: Transmission graduable multimédia. Le problème d'hétérogénéité peut être résolu en combinant le codage de source en couches avec une transmission graduable. Le nœud courant envoie seulement le nombre de couches que les liens peuvent supporter.

d"'apprentissage" permet de connaître les couches à problème. Initialement, le récepteur reçoit la couche de base (A). Lorsque la temporisation expire, le récepteur souscrit à la couche 2 (B). La temporisation est encore courte (aucune congestion n'est détectée). Le niveau de réception atteint ainsi rapidement la couche 3. Le procédé est répété jusqu'à la souscription de la couche 4 où on suppose une congestion (C). La file d'attente augmente et des paquets sont perdus. Une fois que le récepteur détecte une perte de paquets, il diminue son niveau de souscription à la couche 3. La temporisation est alors augmentée (E). Une nouvelle tentative est faite pour obtenir la couche #4. L'échec de la souscription entraine à nouveau une augmentation de la temporisation. Un peu plus tard, une congestion transitoire lors de la réception de la couche 3 oblige le récepteur à diminuer la qualité à la couche 2 (F). La première occurence de cet évenement n'est pas trop préjudiciable à la temporisation et le niveau de souscription revient rapidement à celui de la couche 3.

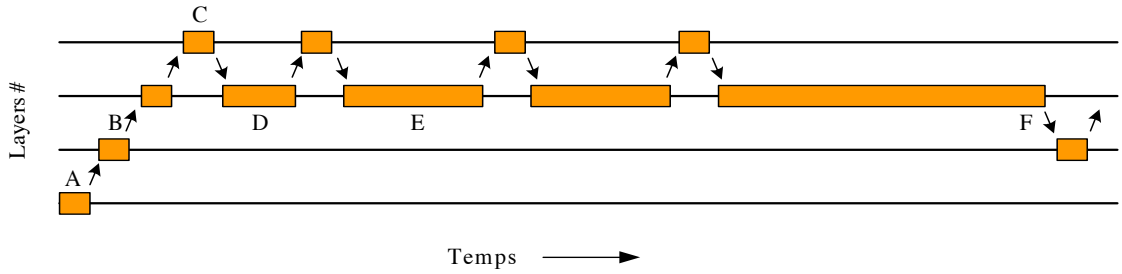


Fig. 2.11: Réactivité temporelle du système RLM. Ce diagramme représente la tratégie de base d'adaptation pour un récepteur donné. Initialement, le récepteur démarre la couche de base puis graduellement ajoute de nouvelles couches jusqu'à la congestion (C). L'effacement d'une couche permet d'ajuster sa souscription à la source.

Ce fonctionnement illustre une bonne réactivité temporelle au problème du transport graduable de la source sans surcharge de la bande passante. Ici, le coût de la protection ne se chiffre pas en termes de redondance mais en termes d'intelligence du réseau à satisfaire chaque récepteur d'un groupe *multicast*. Néanmoins, ce protocole est appliqué à un réseau privilégié puisque le MBone dispose d'une bande-passante satisfaisante (par tunnels) pour un nombre limité de sous-réseaux interconnectés.

2.4 Conclusion

Dans ce chapitre, nous avons présenté un ensemble de solutions conjointes source-canal, pour transmettre un volume d'information graduable dans son intégrité. Selon le niveau hiérarchique où surviennent les pertes, l'effet engendré sur la représentation graduable n'aura pas la même ampleur. Les codages à description multiple proposent donc que la hiérarchie de la source soit cassée lors de son transport. **Cette approche allège fortement le travail du réseau car des descriptions équivalentes de la source sont produites.** Aucun mécanisme de priorité n'est alors nécessaire dans le nœud du réseau, qui fait de son mieux dans ce cas. Cette première solution est parfaitement adaptée à la forme actuelle de l'*Internet Protocol* version 4 où malgré un champ Type of Service (TOS), aucune distinction n'est faite dans l'effacement des paquets.

L'utilisation des codes FEC réalise aussi une description multiple de l'information, mais en employant des techniques classiques de codage canal. Les codes MDS ou $(1+\varepsilon)$MDS sont optimaux en terme de nombre de paquets nécessaires au décodage. La mise en oeuvre dans des serveurs vidéo professionnels témoigne de la fonctionnalité de cette technique. Cette approche offre l'avantage du transport d'un volume important de données [Byers 98] dans de nombreuses descriptions de tailles réduites. Comme pour les codes à description multiple, la taille des descriptions doit être inférieure à la taille maximale des unités de transport (MTU pour Maximal Transport Unit) afin d'éviter toute fragmentation. Ce

mécanisme réseau menace en effet l'équivalence des paquets de données.

Cependant, la forme systématique, dans laquelle les paquets de messages se distinguent des paquets redondants, ne confère pas aux descriptions une même "valeur" dans la complexité du décodage. Cette complexité est en effet linéaire avec le nombre de paquets de messages effacés. En outre, contrairement aux codages à description multiple [Servetto 00], [Goyal 01a], [Jiang 99], les codages FEC sont dans la lignée du principe fondamental de séparation de Shannon en isolant le codage canal du codage source. [Servetto 00], [Goyal 01a], [Jiang 99] préconisent en effet une modification profonde de la source pour qu'elle s'adapte directement aux propriétés du canal.

Une alternative aux deux solutions précédentes est de confier la protection graduable de la source au réseau à son niveau le plus bas. Dans cette autre approche conjointe, le rôle de la source (de l'application) est fortement réduit. Les protocoles de transmission graduable multimédia supposent en effet une gestion de bout en bout de la hiérarchie de la source, et ce principalement aux points d'interconnexion. Cette gestion est particulièrement adaptée pour des réseaux locaux ou privés (LAN, MBone). Mais sa complexité est telle que le déploiement sur des réseaux WAN par DiffServ ou même par IPv6 est aujourd'hui fortement remis en cause.

Chapitre 3

Représentation par transformation Mojette

Sommaire

3.1 Introduction **51**
3.2 Opérateur Mojette-Dirac **52**
3.2.1 Opérateur Mojette direct 52
3.2.2 Supports rectangulaires et condition de reconstruction [Katz 79] 54
3.2.3 Autres supports et théorème de reconstructibilité [Normand 97] 55
3.2.4 Opérateur Mojette inverse 56
3.2.5 Algorithmes et complexité 58
3.3 Utilisation en transmission réseaux **60**
3.3.1 Redondance 61
3.3.2 Reconstruction inexacte 61
3.3.3 Approche multirésolution et transformation Mojette 64
3.4 Conclusion **67**

3.1 Introduction

La transformation Mojette [Guédon 95b][Normand 97] est basée sur la transformation de Radon [Radon 17]. Elle permet de passer d'un signal à deux dimensions à un ensemble de signaux à une dimension. Pour effectuer ce passage, on utilise un mécanisme de sommation suivant des directions déterminées par un couple d'entiers premiers entre eux (p, q). Le résultat donne un ensemble de projections composées chacune d'un ensemble d'éléments somme, appelés bins. La transformation Mojette n'appartient pas aux *problèmes mal posés* comme pour la transformation de Radon. Un nombre fini de projections est en effet nécessaire pour la reconstruction. Cette représentation est en relation directe avec les mécanismes exposés au chapitre précédent. La répartition de l'information dans différentes

projections lui confère en effet des propriétés de description multiple. L'arrivée de chaque projection transmise affine progressivement la reconstruction de l'information initiale.

On insistera dans ce chapitre sur la simplicité de ce mécanisme puisque lors de la mise en œuvre de l'opérateur direct seule l'addition entière sera sollicitée. Pour l'opérateur inverse, c'est la soustraction qui est demandée avec néanmoins une procédure d'initialisation conséquente.

Ce chapitre se limite principalement à une représentation Mojette en prenant comme modèle de luminance d'un pixel une impulsion de Dirac centrée sur sa surface. Elle est néanmoins extensible à d'autres modèles de luminance comme une fonction spline d'ordre 0 (Mojette Haar) ou encore une fonction spline d'ordre 1, modèle pyramidal dont la luminance s'annule à la frontière des pixels voisins. Le premier modèle de Dirac sera développé tant dans sa partie théorique (section 3.2) que dans son utilisation en réseau (section 3.3). Sans que cela ne soit précisé plus avant, c'est cette version qui sera utilisée dans la suite.

3.2 Opérateur Mojette-Dirac

À la différence de la transformation de Radon, l'échantillonnage des projections de la transformation Mojette n'est pas régulier d'une projection à une autre mais est fonction de l'angle de projection tan $\theta = \frac{q}{p}$. On note $\Delta_{pq} = (\sqrt{p^2+q^2})^{-1}$ l'échantillonnage naturel de la projection $proj_{p,q}(m)$ pour une image modélisée par un champ régulier de Dirac de maille carrée unitaire et on écrit la relation entre les modèles continu et discret de l'image par :

$$f(x,y) = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} f(k,l)\delta(x-k)\delta(y-l), \tag{3.1}$$

où $f(k,l)$ représente la valeur de luminance du pixel au point de coordonnées (k,l).

Cette section traite des éléments théoriques de la représentation Mojette (Dirac). Les opérateurs direct et inverse sont successivement décrits. Les conditions de reconstruction sont ensuite définies pour les cas de support rectangulaire et convexe. Enfin, la mise en œuvre de la transformation est décrite et sa complexité détaillée.

3.2.1 Opérateur Mojette direct

Nous établissons ici la définition mathématique d'une projection (p,q) issue de la transformation Mojette directe des échantillons d'image $f(k,l)$. La valeur d'un bin m d'une projection (p,q) est formée par la somme de pixels de l'image $f(k,l)$ se situant sur la droite d'équation $m = -qk + pl$. Pour être accord avec l'écriture de [Guédon 95b] et

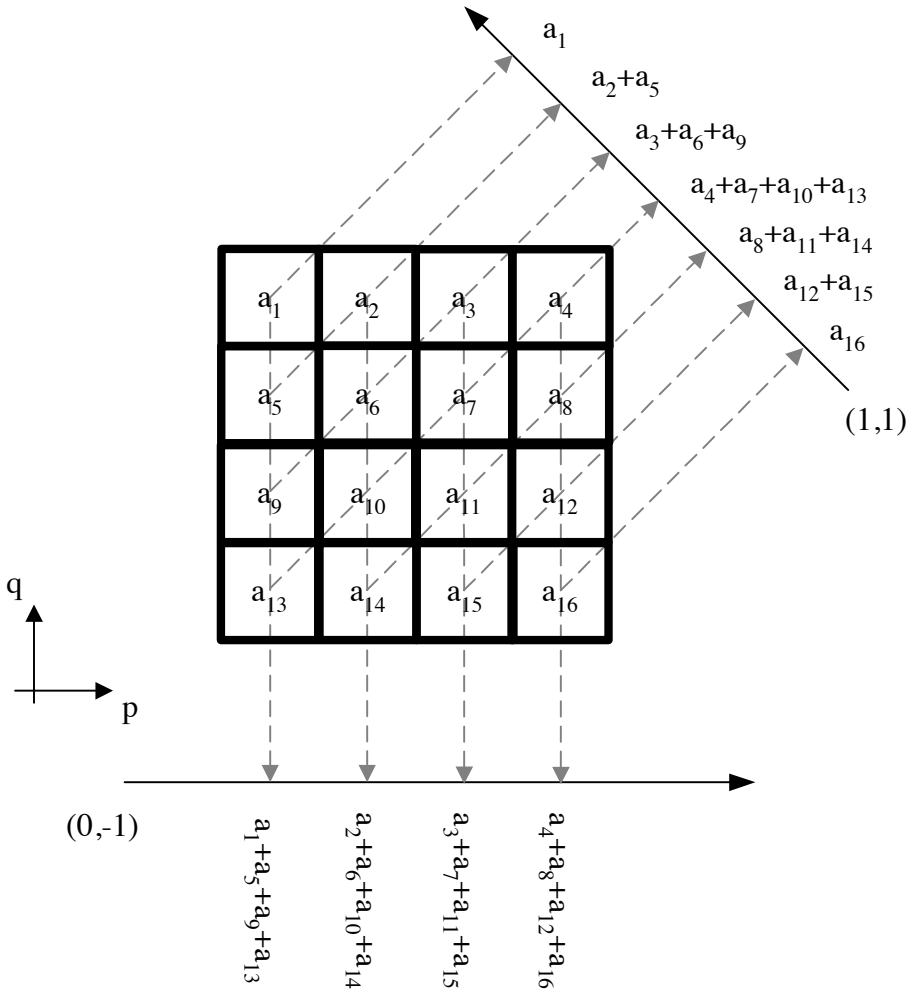


Fig. 3.1: Transformation Mojette-Dirac sur une image 4×4 : projections (1,1) et (0,1).

[Guédon 01b], une projection issue de la transformation Mojette s'écrit :

$$\begin{cases} \mathcal{M}_{p,q} f(k,l) = proj(p,q,m) = proj_{p,q}(m) = proj_{\theta}(m) \\ \mathcal{M}_{p,q} f(k,l) = \sum_k \sum_l f(k,l) \Delta(m + kq - pl) \end{cases} \tag{3.2}$$

où Δ est le symbole de Krönecker[17]. Les directions de projections (p,q) sont toujours choisies telles que p et q soient des entiers premiers entre eux.

Cette équation 3.2 est la définition de la transformation Mojette-Dirac : la somme dans le plan correspond à ne prendre sur chaque ligne que les points de la grille 2D traversés ; c'est un signal 2D échantillonné idéalement (voir figure 3.1). Cette version est la version

[17] $\Delta(m) = \begin{cases} 1 \text{ si } m = 0 \\ 0 \text{ sinon} \end{cases}$.

d'origine de la transformation Mojette dont la complexité algorithmique est la plus réduite.

On définit la transformation Mojette directe des échantillons d'image $f(k,l)$ par $\mathcal{M}f$ qui représente un ensemble de N projections $\mathcal{M}_{p,q}f(k,l)$ tel que

$$\mathcal{M}f = \{proj_{p_i,q_i}, i = 1, 2...N\}. \tag{3.3}$$

La transformation pour deux projections classiques est illustrée figure 3.1.

Cette transformation permet de calculer un nombre important de projections. Dans le domaine continu, la transformation de Radon nécessite en théorie un nombre infini de projections pour retrouver le signal 2D original. Elle se définit donc parmi l'ensemble des *problèmes mal posés.* En revanche, la transformation Mojette est un *problème bien posé.* En effet, les conditions de reconstruction exprimées aux deux paragraphes suivants montre qu'un nombre fini de projections sont nécessaires à la reconstruction de l'image d'origine.

Comme les transformations de Fourier ou de Meyer (ondelettes), la transformation Mojette est linéaire (relation 3.2). Ceci provient de l'opérateur d'intégration présent dans sa définition. Avec la transformation de Radon et la transformation Mojette, on ne cherche pas à obtenir une décomposition orthogonale mais au contraire un frame[18]. C'est la différence fondamentale avec les transformations de Fourier et de Meyer. En effet, ces deux transformations produisent un ordre dans l'importance des coefficients. Pour les transformations de Radon et Mojette, toute projection (un élément dans l'espace transformé) possède le même poids. De plus, chaque élément contient à lui seul la composante continue (DC) (la somme des éléments de la projection), alors que c'est toujours le premier élément pour les transformations orthogonales et qu'il est absent des autres éléments. Cette propriété fondamentale du frame se révèle très intéressante pour la mise en œuvre des transformations directe et inverse : l'ordre de construction ou de reconstruction sera au choix de l'utilisateur et donnera donc un degré de liberté important dans les applications.

3.2.2 Supports rectangulaires et condition de reconstruction [Katz 79]

Nombre de bins sur une projection

Soit un support rectangulaire de $P \times Q$ pixels. La projection d'angle (p,q) possède un nombre de bins donné par la formule :

$$\#bins(P,Q,p,q) = (P-1)\,|q| + (Q-1)\,|p| + 1. \tag{3.4}$$

Ce résultat se trouve aisément en constatant que chacun des termes $(P-1)q$ et $(Q-1)p$ représente les côtés du rectangle vus de l'angle (p,q) et que le 1 supplémentaire permet de passer classiquement du nombre d'intervalles au nombre de points définissant cet intervalle fermé.

[18] Famille génératrice liée d'un espace vectoriel [Chou 99].

Critère de Katz

Le lemme de Myron Katz donne le résultat suivant.

Pour un support R rectangulaire de $P \times Q$ pixels et un ensemble S de N projections $\{(p_i, q_i), i \in N\}$, on a l'équivalence :

$$S \text{ reconstruit } R \Longleftrightarrow P_N = \sum_i |p_i| \geq P \text{ ou } Q_N = \sum_i |q_i| \geq Q. \tag{3.5}$$

Ce résultat capital possède une visualisation vectorielle simple qui sera utilisée par la suite. Chaque angle de projection (p, q) peut être vu comme représenté par un vecteur de composantes p et q. L'équation 3.5 traduit le fait que la somme desvaleurs absolues des différents vecteurs de projections est un vecteur de composante P_N et Q_N. Et, lorsque ce vecteur ne rentre plus dans le rectangle $P \times Q$ alors la reconstruction est possible.

3.2.3 Autres supports et théorème de reconstructibilité [Normand 97]

Nombre de bins sur une projection

Pour un support non rectangulaire, l'équation 3.4 n'est plus applicable. Les côtés d'un polytope quelconque vus de l'angle (p, q) ne s'écrivent pas toujours aussi aisément. Aussi, on choisit pour déterminer le nombre de bins d'utiliser la différence entre les indices m_1 et m_2 du premier et du dernier bin. On a donc

$$\#bins(m_1, m_2) = |m_2 - m_1| + 1. \tag{3.6}$$

Cette simple observation nous permettra de dénombrer les bins dans le cas de supports variées présentés aux chapitres 4 et 5.

Liens entre la transformation Mojette et la Morphologie Mathématique

On associe à une projection d'angle (p, q) un élément structurant à deux pixels (ÉS2P) sous la forme d'un couple de points $\{0, (p, q)\}$.

Concernant la transformation Mojette, la rétroprojection de tous les bins d'une même projection en correspondance directe avec un pixel revient à effectuer l'ouverture morphologique (érosion puis dilatation) du support avec l'élément structurant correspondant à la direction de projection. La figure 3.2 illustre cette opération.

Ainsi, la reconstruction d'une image par un ensemble suffisant de projections s'apparente pour le support à une suite d'ouvertures par les ÉS2P correspondants aux vecteurs de projections. On écrit alors le critère de reconstructibilité de Katz (Éq. 3.5) sous la forme d'un critère ne faisant intervenir que des dilatations. Une image convexe n'est reconstructible au sens de la transformation Mojette inverse que si le résultat des dilatations successives des ÉS2P correspondants aux directions de projection ne peut être inclus dans

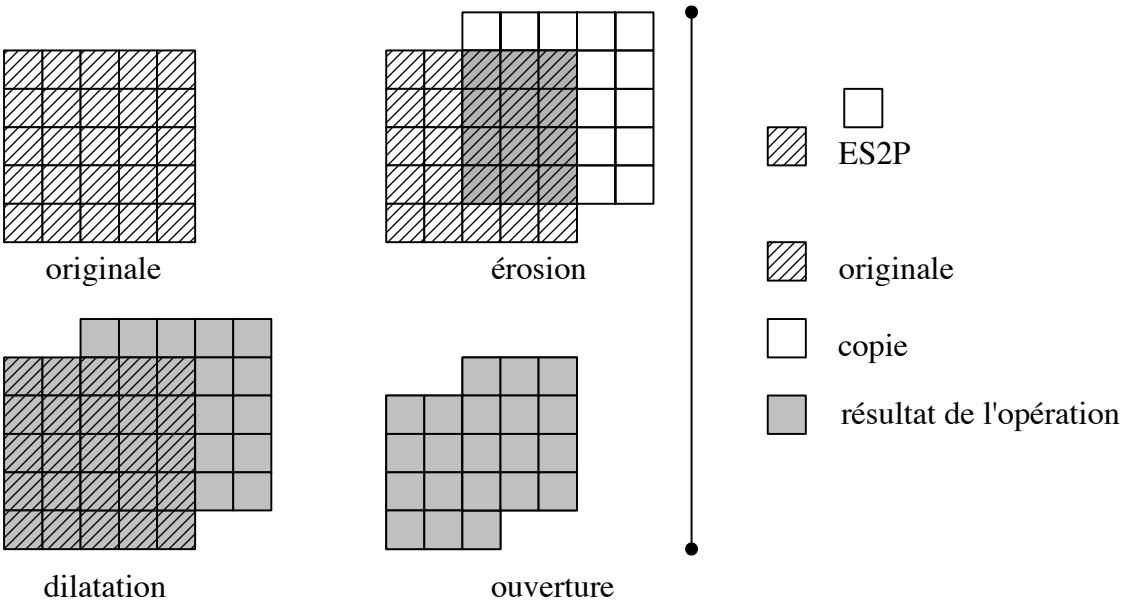


Fig. 3.2: Ouverture d'un carré par un élément structurant à deux pixels (ÉS2P).

le support de l'image (ne serait-ce que d'un pixel). La figure 3.3 montre la construction de la plus petite image non reconstructible par une série de dilatations ainsi qu'un exemple de support convexe[19] reconstructible au sens de la Mojette inverse.

Théorème de reconstructibilité [Normand 97]

Le résultat général s'exprime également dans le théorème suivant par l'équivalence des deux propositions.

Énoncé 1 : les deux propositions suivantes sont équivalentes

i) $f(k,l)$ définie sur le convexe G est reconstructible par $\{proj_{p_i,q_i}, i = 1,2...N\}$;

ii) R construite par la suite des N dilatations $\{(O,(p_i,q_i), i = 1,2...N\}$ n'est pas inclus dans G.

Deux propositions peuvent aussi être exprimée en d'autres termes morphologiques :

Énoncé 2 : les deux propositions suivantes sont équivalentes

i) G est reconstructible par $\{(O,(p_i,q_i), i = 1,2...N\}$;

ii) l'érosion de G par R est l'ensemble vide.

3.2.4 Opérateur Mojette inverse

Dans le procédé de reconstruction, on cherche sur chaque projection les bins qui sont en correspondance bi-univoque avec un pixel de l'image. Ce sont en général les pixels de bord

[19] Un ensemble de points $\{(p_i, q_i)\}$-convexe est tel que chaque ligne d'orientation (p_i, q_i) avec $i \in N$ est au plus formée d'un segment discret. La pq-convexité est ici succintement définie. Les travaux de thèse de Nicolas Normand développent cet aspect dans [Normand 97].

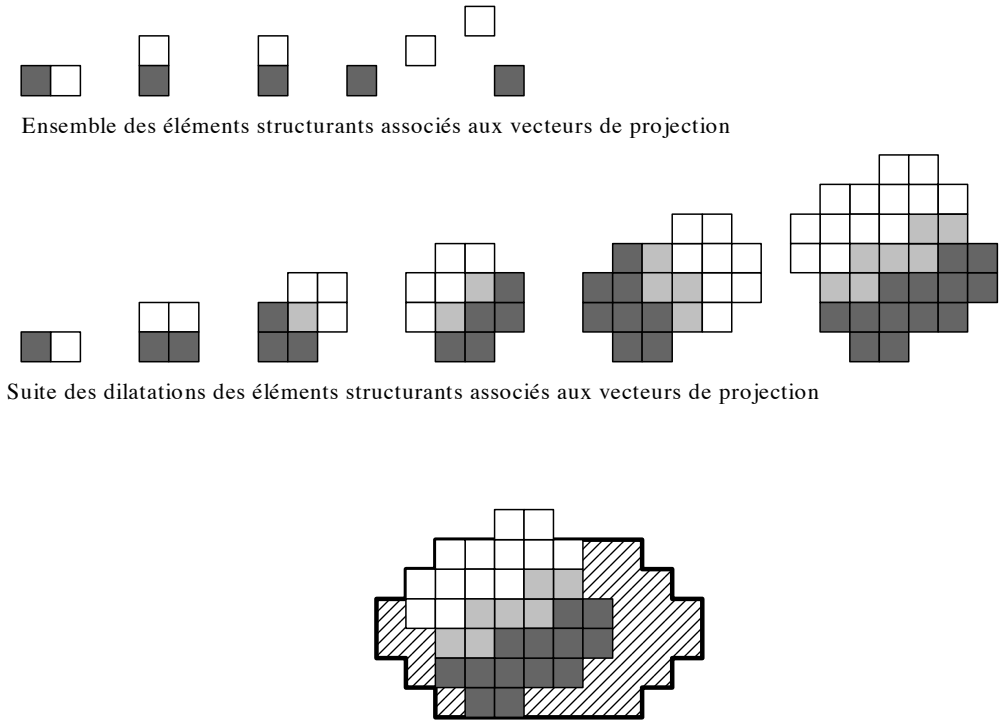


Fig. 3.3: Construction par dilatation de la plus petite image non reconstructible.

de l'image. L'objectif de cette correspondance est d'obtenir un algorithme qui ne dépende que du nombre de pixels à reconstruire et du nombre de projections présentes dans le jeu de projections utilisé car la mise à jour du système va engendrer une mise à jour par projection (à l'endroit où se projetait le pixel que l'on vient de reconstruire). On rétro-projette alors dans l'image la valeur d'un bin candidat. Une fois ce pixel reconstruit, on met à jour la valeur des bins des autres projections qui résultent de sa contribution. Cela concerne un et un seul bin par projection. On procède ainsi itérativement jusqu'à la reconstruction complète de l'image de départ. Cette méthode de reconstruction s'apparente à la résolution d'un système linéaire triangulaire. La figure 3.4 montre le procédé de reconstruction sur une image (4×4). À la fin de la reconstruction, chaque projection doit posséder des valeurs de bins identiques et toutes nulles. Toute l'information contenue dans les projections a servi à reconstruire l'image de départ.

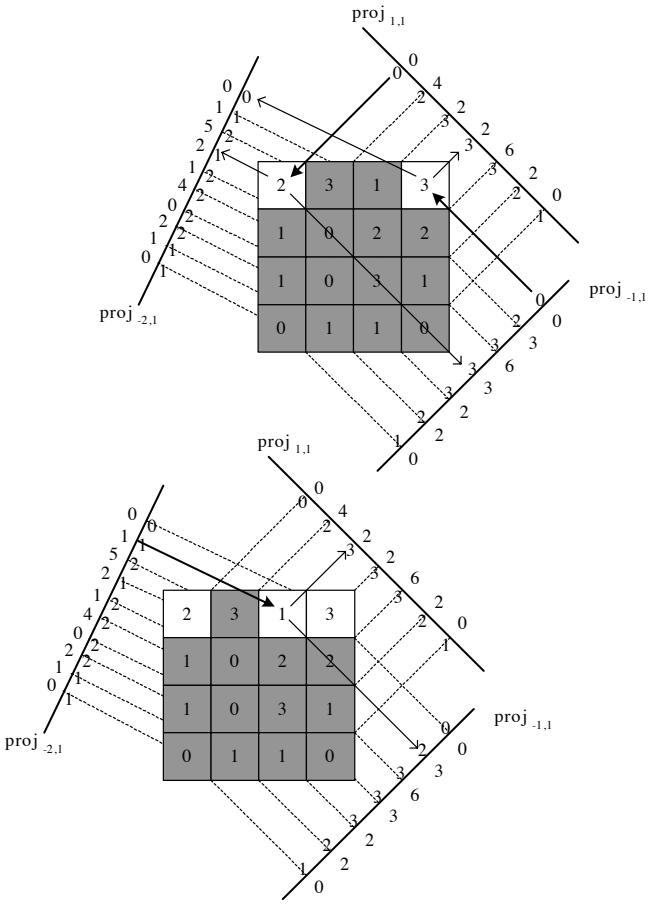


Fig. 3.4: Exemple de transformation Mojette inverse : support 4×4 reconstruit à partir des trois projections {(1,1) ;(-1,1) ;(-2,1)}.

3.2.5 Algorithmes et complexité

Opérateur direct

Débutons par le cas le plus simple. Prenons un support rectangulaire $(P \times Q)$ et une projection (p, q). Son calcul est donné par Algo_Moj1 :

Algo_Moj1

Début

Pour chaque lig de 1 à Q faire

Pour chaque col de 1 à P faire

m ⟵ p.lig − q.col *//calcul du bin m correspondant*

proj(m) +⟵ image(lig, col) *//incrémentation du bin*

FinPour

FinPour

Fin

Puisque chaque pixel contribue à un bin, la complexité est nécessairement de $\mathcal{O}(PQ = I)$ pour une projection quelconque. Si on veut maintenant calculer un ensemble de N projections, la complexité est de $\mathcal{O}(IN)$, linéaire dans le nombre de pixels et dans le nombre de projections.

Dès lors que le support n'est plus rectangulaire, on peut s'attendre à un ajout de complexité pour décrire la forme pixel par pixel. Évidemment, si on inscrit la forme (quelconque) dans un support rectangulaire en mettant à zéro les pixels additionnels, on ne changera pas l'ordre de complexité précédent ni le comportement de l'opérateur Mojette. Ceci est un avantage sur les autres transformations comme Fourier par exemple : border le signal par des valeurs nulles ne change pas la transformée.

Supposons que nous connaissons la fonction f et son inverse permettant de passer d'un dénombrement des pixels de la forme d'indices 2D en indices 1D.

Par exemple pour l'algorithme précédent :

$$compteur = f(col + (lig - 1)P)$$

et

$$f^{-1}(compteur) = (col, lig)$$

en utilisant *div* et *mod*. Pour les I pixels de la forme à projeter, on a alors l'algorithme

Algo_Moj1_convexe
Début
  Pour compteur de 1 à $|$ *faire*
    (lig, col)$\longleftarrow f^{-1}$*(compteur)*
    m $\longleftarrow$ *p.lig − q.col* *//calcul du bin m correspondant*
    proj(m) $+\longleftarrow$ *image(lig, col)* *//incrémentation du bin*
  FinPour
Fin.

On se rend donc compte qu'il n'y a pas d'ajout de complexité (la fonction f peut toujours être tabulée). D'autre part, rien n'indique ici une propriété de convexité. En fait, la forme est vraiment quelconque.

Opérateur inverse

La mise en œuvre "visuelle" de la transformation Mojette inverse procède comme suit. On part d'une correspondance univoque entre un point du support et un point d'une

projection, on reconstruit le point du support et on parcourt toutes les projections pour soustraire la valeur du correspondant à ce bin. En pratique cela pose deux problèmes. Le premier est de trouver une correspondance univoque sans affecter la complexité de l'algorithme. Le second est de retrouver la position d'un élément du support à partir de la position d'un bin et ce, toujours sans affecter la complexité.
On décompose en phase 1,2 et 3 respectivement la recherche de correspondance univoque, la rétroprojection et la mise à jour des projections.
Pour la phase 1, une façon de faire est de définir pour chaque pixel une liste des bins correspondants pour chaque projection. Autrement dit, un pixel comprend à la fois une valeur (initialisée à -1) et un ensemble de N valeurs indiquant la position du bin m_i de la projection i sur lequel est projeté le pixel. On gère alors une liste de pixels en correspondance univoque et la mise à jour de cette liste lors de la reconstruction d'un pixel est automatique : on enlève le pixel que l'on reconstruit et on ajoute (éventuellement) de nouveaux pixels qui sont passés en correspondance univoque. Cette fois-ci, la complexité associée est bien $\mathcal{O}(1)$.
En phase 2, le problème est le suivant : on dispose de la position d'un bin dans une projection (i, m) et on désire savoir sur le segment discret D quel est le seul pixel de coordonnées (lig, col) qui n'a pas été encore reconstruit. La méthode de l'indice unique évite le problème avec une complexité $\mathcal{O}(1)$. Chaque pixel de l'image de départ est numéroté. Pour une image $P \times Q$, le pixel (lig, col) possède le numéro $compteur = (lig - 1)P + col$. À chaque bin, on associe la valeur (somme des pixels), la valence (nombre de pixels) et une position (la somme des numéros). Lorsqu'un pixel est en correspondance univoque sa position donne le numéro du pixel. Lorsque l'on met à jour un bin, on soustrait la position du pixel. La valence et la position du pixel sont obtenues en initialisation de l'opérateur inverse par transformation directe du support de l'image contenant respectivement des 1 et les numéros aux pixels de la forme.
Seule la phase 3 demande $\mathcal{O}(N)$. Les deux premières phases sont simplifiées. Au total, l'algorithme de complexité où N est le nombre de projections et I le nombre de pixels est donné par Algo_MojInv (Annexe 2). Il est aisé de voir que l'on a rajouté une tâche importante en initialisations diverses : la complexité de cette phase est $\mathcal{O}(4.IN)$.

3.3 Utilisation en transmission réseaux

La transformation Mojette propose de répartir l'information initiale dans un ensemble plus ou moins redondant de projections. Cet ensemble de signaux 1D est assimilable aux paquets (PDU - Protocol Data Unit) qui transitent sur les réseaux. Nous allon voir dans cette section comment cet opérateur fournit une protection des données utilisateur et réalise une mise en charge naturelle des unités de transport.

3.3.1 Redondance

La représentation par transformation Mojette est redondante par la répartition des éléments d'information initiaux dans un plus grand nombre de bins (symboles transmis). On définit le taux de redondance, Red, comme le simple rapport entre le nombre de pixels et le nombre de bins moins un. Soit Red cet index, on a

$$Red = \frac{\#bins}{\#pixels} - 1. \tag{3.7}$$

L'ajout dans cette équation du décalage d'une unité provient de la façon classique de calculer la redondance dans le monde des réseaux.

La redondance est nulle lorsque le nombre de bins est égal au nombre de pixels comme c'est le cas avec une projection dégénérée du type $(Q, 1)$. Dès que l'on choisit un ensemble non-dégénéré de projections qui permettent une reconstruction, on a $Red > 0$. Il faut remarquer que même dans le cas où $Red > 0$, la reconstruction n'est pas forcément possible. Par un exemple, un support 3×3 représenté par les projections (1,1) et (-1,1) conduit à 9 pixels et 10 bins soit $Red = \frac{1}{9}$ mais la reconstruction totale est impossible (5 pixels seulement sont reconstruits).

Par le théorème de reconstructibilité, [Normand 97] a donné un moyen de savoir si un ensemble de projections suffisait à la reconstruction d'une image de support convexe donné. Un tel ensemble est dénommé ensemble suffisant de projections (i.e. pour reconstruire une image donnée). Il est entendu que l'on peut définir autant de projections que souhaité, ce qui permettra ensuite d'introduire de la redondance dans la transmission de la source. La question est de savoir si l'on pourra reconstruire l'image avec un sous-ensemble de l'ensemble initial des projections en supposant par exemple l'effacement d'une des projections. Exprimé différemment, il s'agit de déterminer si à partir d'un ensemble suffisant de projections, on peut trouver un ensemble redondant tel que si on le prive d'un élément quelconque, il forme un sous-ensemble suffisant. Le corollaire du théorème de reconstructibilité assure qu'un tel ensemble existe.

Corollaire du théorème de reconstructibilité

Soit un ensemble de vecteurs de projection $\{(p_i, q_i)\}_{i\in(1,M]}$ et une image reconstructible $f(k, l)$, on peut toujours trouver un vecteur de projection supplémentaire (p_{M+1}, q_{M+1}) tel que l'image peut être reconstruite avec n'importe quel sous-ensemble de $M-1$ projections issues des M initiales et la projection supplémentaire.

Les chapitres 4 et 5 vont dans ce sens et déterminent les angles de projections supplémentaires pour un support convexe.

3.3.2 Reconstruction inexacte

Ce paragraphe répond à la question suivante : comment procède-t-on dès lors que les conditions de reconstructibilité ne sont pas satisfaites ? Il résume le résultat essentiel

obtenu dans la thèse d'Olivier Philippé [Philippé 98].

Nous cherchons ici à obtenir une reconstruction inexacte contrairement à la reconstruction exacte quand les conditions de reconstructibilité sont satisfaites (critère de Katz, morphologie mathématique). Nous avons une image I à reconstruire à partir d'un ensemble de N projections S_N. Lorsque des projections sont manquantes, un très grand nombre de reconstructions est possible. Cependant si les éléments d'information produitent par la source sont sous contrainte (corrélation d'une image), une décomposition unique peut être utilisée à partir de vecteurs de bases dénotés fantôme pour entamer une reconstruction inexacte. Un fantôme élémentaire (figure 3.6) est constitué par la suite des convolutions d'ÉS2P valués par +1 (à l'origine) et -1 (sur le pixel (p,q)) (figure 3.5), ce qui donne zéro pour la somme dans la direction de projection d'où le nom de fantôme.

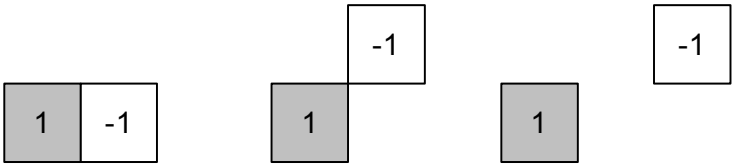


Fig. 3.5: ÉS2P valué pour $S_3 = \{(1,0),(1,1),(2,1)\}$.

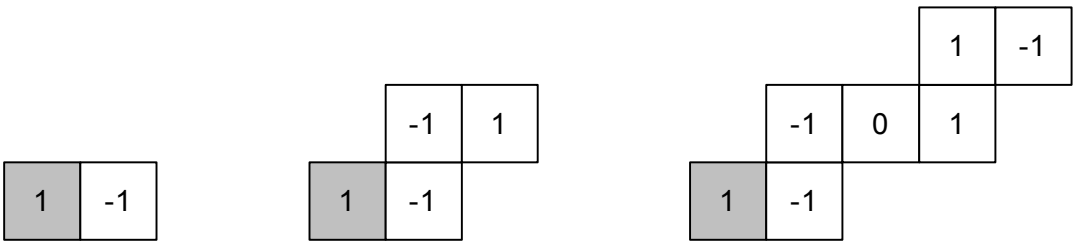


Fig. 3.6: Fantôme élémentaire pour $S_3 = \{(1,0),(-1,1), (2,1)\}$.

Pour une reconstruction inexacte, le nombre d'inconnues est le cardinal de l'ensemble érodé par les projections dont on dispose. On note r ce nombre d'inconnues. On place alors le fantôme élémentaire f_i ou toute combinaison linéaire de celui-ci sur chaque pixel de l'érodé de sorte à obtenir à chaque fois une forme contenue dans la région de départ et un vecteur de base de l'espace nul (un seul pixel de l'érodé est à 1 et les autres sont à zéro). On note I_{sc} la région reconstruite par N projections sous la contrainte d'un érodé nul (soit r pixels nuls), et a_i la valeur inconnue du pixel i de l'érodé. Olivier Phillipé démontre qu'il est possible de décrire l'image de départ comme la somme de l'image sous-contrainte et de sa décomposition sur l'espace projeté en mettant l'érodé à zéro, soit :

$$I = I_{sc} + \sum_{i=1}^{r} a_i f_i. \tag{3.8}$$

Nous illustrons cette décomposition sur une image 4×4 projetée en deux angles (1,0) et (-1,1). En utilisant alors la morphologie mathématique, l'érosion du support par les ÉS2P {(O, (-1,1)) ; (O,(1,0)} est constitué d'une région (non vide dans le cas présent) composée de six pixels nommés a_1 à a_6. La région érodée (en gris sombre) est représentée figure 3.7.

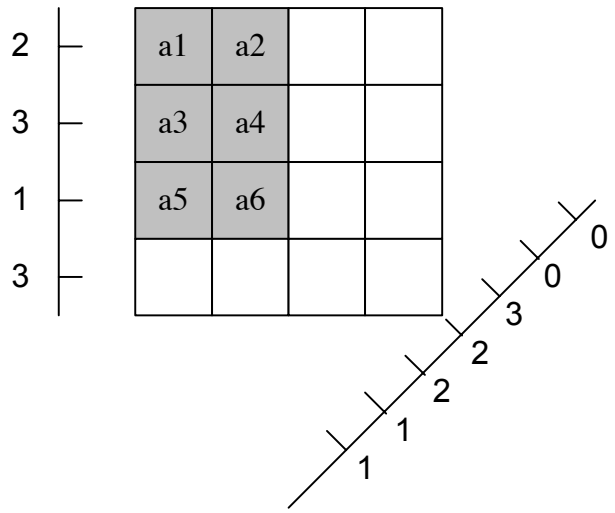


Fig. 3.7: Région érodée par les projections (-1,1) et (1,0).

Si les six valeurs des pixels sont initialisées, une reconstruction unique peut être obtenue à partir des deux projections disponibles. Le premier terme de la somme sur la figure 3.8 nous donne cette reconstruction pour des pixels de l'érodé forcés à zéro. Deux pixels sont reconstruits exactement. Hormis l'érodé, les autres pixels (gris clair) sont déduits de l'opérateur Mojette inverse. Les autres termes de la somme sont les compositions des fantômes associés aux projections (-1,0) et (1,0). Ces motifs multipliés par une constante α_i ne modifient pas les deux projections et ce, quelle que soit leur position sur l'image. Les fantômes sont cependant centrés sur chaque pixel de l'érodé et n'affectent pas les pixels reconstruits. Des contraintes externes peuvent être appliquées à chaque pixel. Dans le cas de pixels corrélés (pour une image), les corrélations voisines sont utilisées de manière itérative.

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 5 & -2 \\ 0 & 0 & 3 & -2 \\ 1 & 1 & 2 & -1 \end{bmatrix} +$$

$$\alpha_1 \begin{bmatrix} + & - & 0 & 0 \\ 0 & - & + & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 & + & - & 0 \\ 0 & 0 & - & + \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 & 0 & 0 & 0 \\ + & - & 0 & 0 \\ 0 & - & + & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} +$$

$$\alpha_1 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & + & - & 0 \\ 0 & 0 & - & + \\ 0 & 0 & 0 & 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ + & - & 0 & 0 \\ 0 & - & + & 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & + & - & 0 \\ 0 & 0 & - & + \end{bmatrix} .$$

Fig. 3.8: Décomposition unique par les fantômes de base issus des projections (-1,1) et (1,0).

3.3.3 Approche multirésolution et transformation Mojette

Dans le chapitre 5 de la thèse d'Olivier Phillipé[Philippé 98], un schéma multirésolution à base de transformations en ondelettes est proposé dans le cadre de transmission sur réseaux ATM. La représentation multirésolution de l'image est décrite comme étant équivalente à la représentation multirésolution des projections. Cependant ce schéma est limité à un choix restreint de projections[20] où seul les filtrages "horizontal" et "vertical"

[20] Pour effectuer j décomposition et garder la possibilité de reconstruction, il faut choisir un ensemble de projections ayant la propriété suivante :

$\forall i \in \{1, ...M\}, \exists j \in N^*, \exists k_i \in Z | \begin{cases} p_i = 2^j k_i \\ q_i = 1 \end{cases}$.

Un résultat similaire peut être obtenu pour une version de basse résolution verticale en choisissant $q_i = 2^j k_i$ et $p_i = 1$.

sont autorisés. Il ne correspond en fait à une représentation multirésolution de l'image que lorsque le nombre de bins sur chaque projection et les dimensions de l'image sont pairs. Seule l'ondelette de Haar est utilisable. La figure.3.9 illustre ce propos pour une image de dimension 512×512, découpée en blocs de taille 32×32. Pour chaque bloc, on effectue la transformation Mojette sur un ensemble reconstructible de quatre projections. Les projections sont ensuite filtrées par l'ondelette de Haar. La transformation Mojette est réalisée à partir de ces projections filtrées. La reconstruction et l'image différence sont représentées figure 3.9.

Fig. 3.9: Reconstruction de l'image Lena et image différence après $j = 1$ filtrage de Haar sur chaque projection de l'ensemble $S = \{(16, 1); (8, 1); (4, 1); (-4, 1)\}$.

Dans l'objectif d'utiliser des ondelettes beaucoup plus puissantes, la transformation Mojette est envisagée après l'analyse multirésolution de l'image. Ce faisant, on sécurise les différentes bandes de fréquences spatiales comme le montre la figure 3.10.

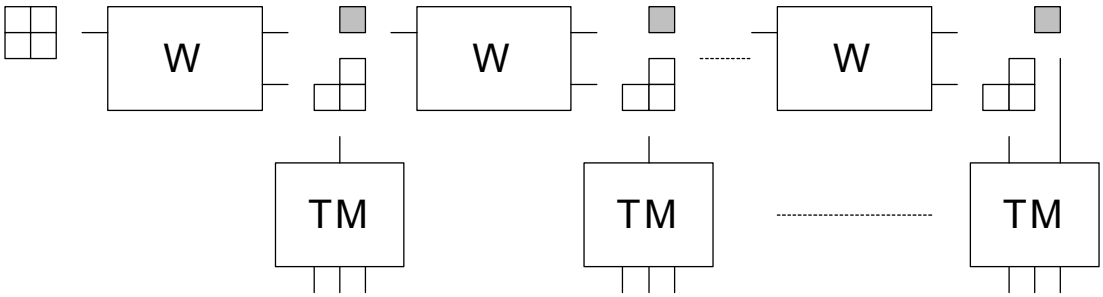


Fig. 3.10: Analyse ondelette (W : Wavelet) d'une image et transformation Mojette (TM) sur chaque bande de fréquences spatiales.

Applications

Supposons que l'on décompose le signal image en blocs. Chaque bloc est alors représenté globalement par une série de $M + 1$ projections. L'effet d'une perte isolée est assimilée à la perte d'une projection sur un bloc. On est capable de reconstruire le signal à l'aide des M projections reçues. Dans le cas d'une congestion, plusieurs cellules sont perdues. En regroupant les projections de même direction, on effectue un multiplexage temporel. L'effet de ce multiplexage est de répartir la perte d'une projection sur plusieurs blocs.

Ce type d'application peut être associé à une transmission sur un ou plusieurs canaux. En effet, en transmettant une direction de projections par canal, on garantit une indépendance accrue des différentes projections, qui peuvent être utilisées de manière asynchrones. La reconstruction peut en effet débuter dès l'arrivée d'une projection en initialisant la valeur des bins des autres projections à zéros comme le montre la figure 3.12. Cependant, si la théorie permet d'envisager une telle approche, l'aspect du codage canal la rend hasardeuse. La mise en œuvre s'avérerait coûteuse, tant par le nombre de canaux utilisés que par la complexité des algorithmes de synchronisation qu'elle nécessite. Dès lors, l'application est réduite à la transmission sur deux canaux ATM d'une version basse et haute résolution de l'image, ce qui, en termes théoriques, est décrit dans le problème d'origine de la description multiple. La figure 3.11 montre comment découper les différentes bandes de fréquences de façon à être en adéquation avec ce type de transmission. Le paramètre red permet de gérer la redondance des projections sur chaque bande de fréquences alors que le paramètre β est un indice de regroupement similaire à celui du nombre de coefficients d'une transformée DCT. On regroupe ainsi les projections de β bandes de basses fréquences pour les envoyer sur un canal temps réel de type CBR[21] ou VBR-rt[22] pour ATM. Les autres projections sont envoyées sur un canal moins coûteux de type VBR-nrt ou ABR[23]. De cette façon, la qualité de service offerte par le protocole ATM permet de reconstruire une version basse résolution de l'image en temps réel. Une version haute résolution peut être obtenue en différé.

[21] Constant Bit Rate
[22] Variable Bit Rate - real time
[23] Available Bit Rate

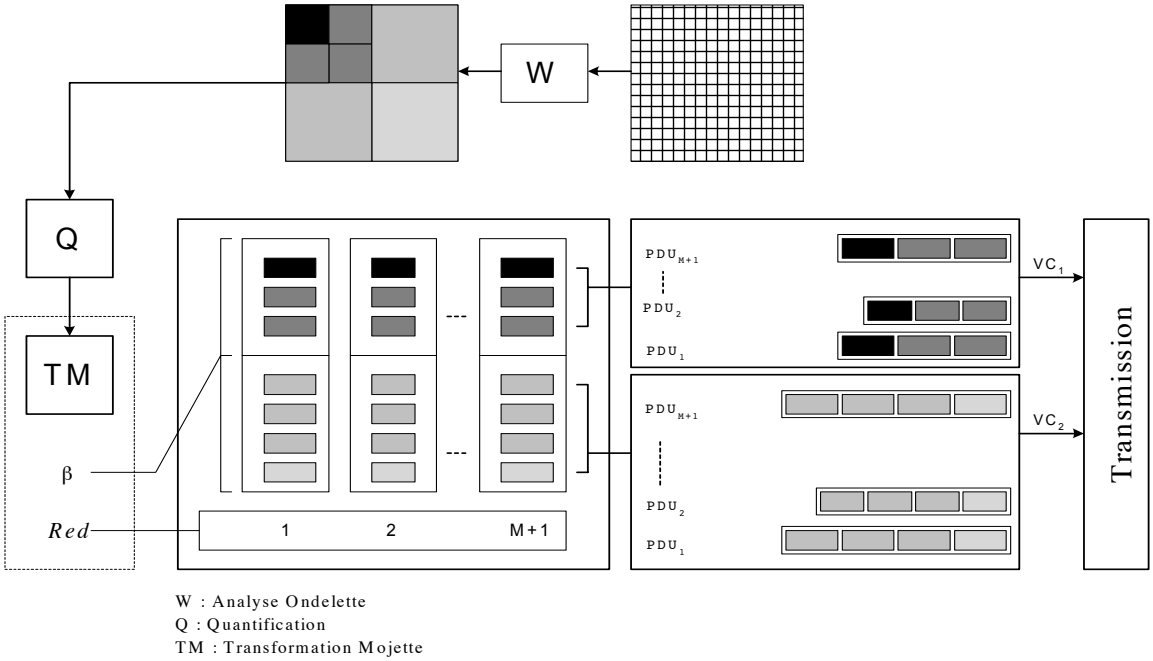


Fig. 3.11: Transmission multi-résolution et multi-canaux.

3.4 Conclusion

Dans ce chapitre, nous avons apporté les éléments théoriques qui constituent l'outil Mojette. Les conditions d'inversion par le critère de Katz d'une part et en termes morphologiques d'autres part ont été formulées. La mise en œuvre des opérateurs direct et inverse sont détaillées tant dans son écriture algorithmique que dans l'évaluation de sa complexité.

L'utilisation en réseau semble *a posteriori* naturelle[24] [Guédon 95a]. Les projections engendrées sont assimilables aux unités de transport qui transitent sur les réseaux d'aujourd'hui (IP ou ATM). La description redondante d'une image contribue à la rendre robuste au cours de la transmission. En effet, les regroupements par angles de projections effectuent un entrelacement des données initiales pour homogénéiser les pertes, qui survenues de manière trop importante, peuvent être compensées par la décomposition unique en fantômes de base (§ 3.3.2). De plus, la reconstruction peut être asynchrone (figure 3.12) ce qui marque un avantage majeur vis-à-vis des codes correcteurs classique de type MDS (§ 2.1) avec lesquels la transformation Mojette entre en concurrence directe.

La forme rectangulaire n'est pas forcément adaptée à tout (e.g. description polygonale d'une image). Pour chaque polygone, on prend quelques projections au lieu de prendre le rectangle englobant. Dès lors, on ne conserve de la forme que la partie "utile" en ôtant

[24] L'utilisation en tatouages d'images [Autrusseau 01] et en analyse d'un contenu vidéo n'est pas détaillée ici.

les coins sans changer la transformation. Cette facilité nous permettra d'employer des supports convexes dans les chapitres à venir. On exploite alors la relation intrinsèque qui existe entre la forme et les angles de projections pour la protection de données sources.

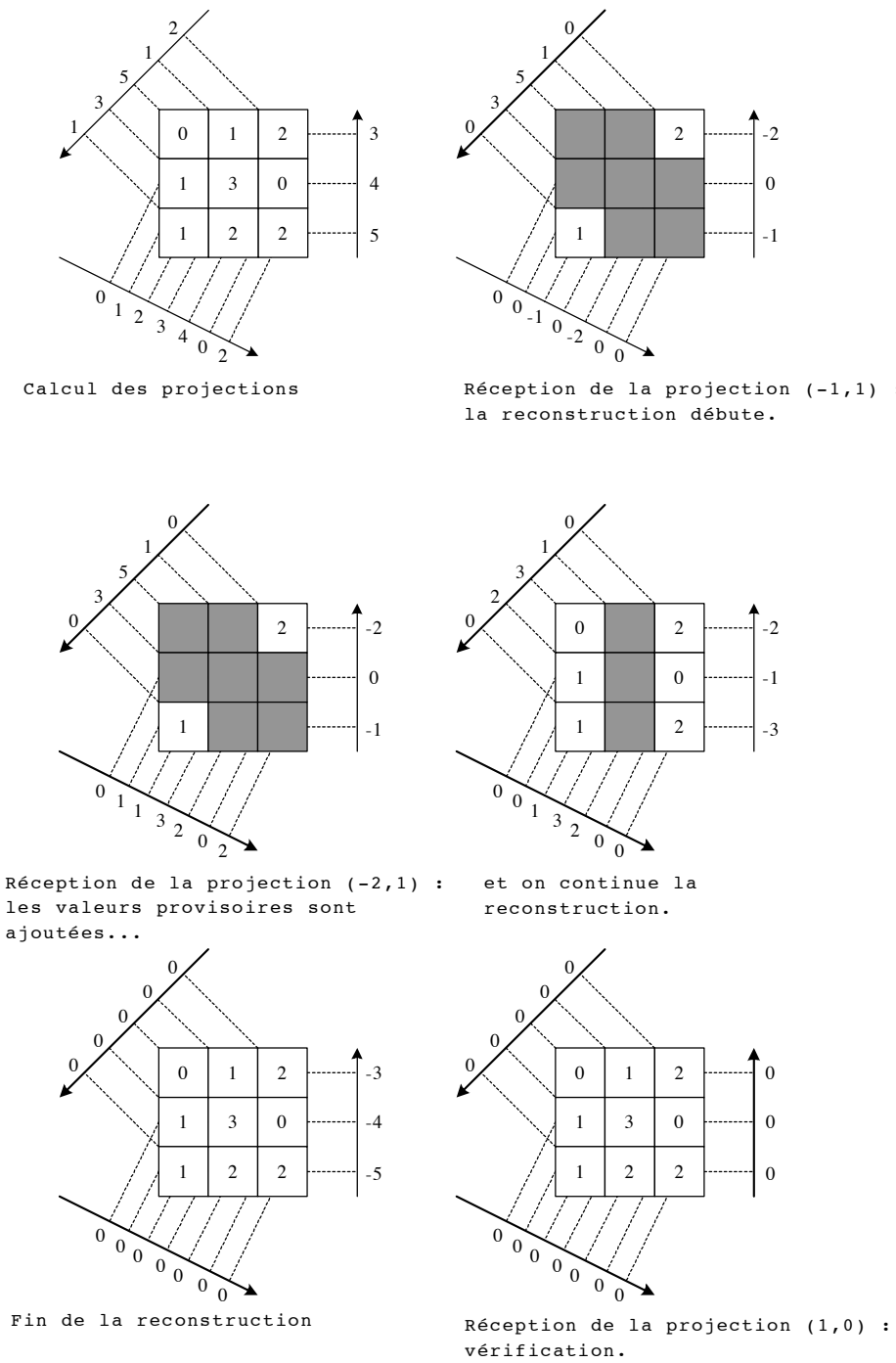


Fig. 3.12: Reconstruction asynchrone par la transformation Mojette inverse. L'algorithme peut débuter à l'arrivée de la première projection (-1,1) en initialisant la valeur des bins des autres projections (1,0) et (-2,1) à zéros. La contribution des éléments reconstruits est retiré des projections que celles-ci soient ou ne soient pas reçues.

Conclusion de la première partie

Cette première partie synthétisait une étude bibliographique sur les codages de source et de canal. Dans le chapitre 1, nous avons présenté séparément ces deux codages au travers d'exemples significatifs de mise en œuvre. De manière générale, la source est soumise tout d'abord à une transformation orthogonale afin d'extraire les corrélations existantes entre ses éléments. Le regroupement des éléments de la transformée (fonction de l'énergie du signal) et leur quantification éventuelle amènent à la compression du signal. Le flux binaire obtenu possède alors une structure très hiérarchisée. Un codage de canal est ensuite réalisé comme protection aux erreurs pouvant survenir au cours de la transmission. Qu'il se traduise par un code correcteur indépendant du média qu'il protège ou bien par un mécanisme réseau visant à gérer la graduabilité de la source, le contrôle d'intégrité entraîne inévitablement un sur-débit. L'optimisation séparée du codage de source et du codage de canal débouche donc sur des performances en compression en augmentant la sensibilité de la source aux pertes d'information et à un ajout uniforme de redondance côté codage de canal pour compenser les effets de perte. Le codage conjoint source/canal évite ce paradoxe en effectuant un équilibrage des rendements des codages de canal et de source en accord avec les caractéristiques du canal (délai de transmission, perte de paquets, gigues...). Dans ce contexte, le codage conjoint source/canal se doit de préserver l'indépendance de la source vis-à-vis du réseau.

La description multiple, dans ce cas, réalise ce principe dans le cadre de transmission sur canal "best effort", le canal fait ce qu'il peut et ne gère aucune forme de hiérarchie de la source. Le mécanisme décrit dans le chapitre 2 fait état d'une rupture de la hiérarchie de la source et d'une répartition du volume initial d'information dans de multiples descriptions toutes équivalentes. La structure engendrée (description de la source et redondance) est autonome et ne donne pas lieu, en cas de perte de descriptions, à une retransmission préjudiciable à des contraintes de temps réel. Une version dégradée de la source est alors obtenue.

Un opérateur à description multiple est également présenté. La transformation Mojette est basée sur la transformation de Radon mais en version complètement discrète. Elle réalise des projections de l'information dans le sens direct et reconstruit les données initiales lors de l'inversion en n'utilisant que des opérations d'addition et de soustraction. La transformation inverse n'est pas un problème mal posé comme pour la transformation de Radon.

Ici, un nombre fini de projections est nécessaire pour l'inversion et la reconstruction est exacte. Des conditions de reconstructibilité sont rappelées en termes de morphologie mathématique. Les projections générées, signaux à une dimension, sont analogues à des unités de transport motivant l'utilisation de la représentation de la transformation Mojette sur des protocoles de transport de paquets. En banalisant ces projections (qui ont une même importance dans la reconstruction), la transformation Mojette est un code à description multiple de complexité réduite.

L'application du codage à description multiple sur des sources audio, image et vidéo prouve la maturité de ce mécanisme de compensation de pertes. Cependant les améliorations apportées par ce codage ne doivent pas aller à l'encontre de l'évolution des sources actuelles. Les schémas classiques de description multiple impliquent la redéfinition du codage de source par rapport au canal de communication (qu'il soit dédié au stockage ou à la transmission). Le schéma de transmission multirésolution de la figure 3.11, page 67 ne propose pas de solution réelle de quantification pour les coefficients ondelettes. En remplaçant la transformation Mojette par le quantificateur scalaire à description multiple (MDSQ) de Vinay Vaishampayan [Vaishampayan 93], on peut obtenir une forme non redondante de codage [Servetto 00]. On a modifié dans ce cas le codage de source à des fins de description multiple. Néanmoins, cette forme de codage ne semble pas assez robuste étant données les pertes observées sur un canal de l'Internet (figures 1.7 à 1.10, page 22). L'ajout de codes correcteurs[25] au schéma proposé dans [Servetto 00] témoigne de la sensibilité du codeur aux pertes d'information.
Le schéma de transmission multirésolution proposé par Olivier Phillipé suppose, en outre, une qualité de service distincte pour chaque canal mis en jeu. Il ne s'applique dons pas à des communications sur canal "best effort". Or, dans la suite, c'est ce type de canal qui est envisagé. L'agrégation des flux dans le cas d'une structure graduable de la source dans des descriptions équivalentes est alors nécessaire pour ce type de transmission. Aussi, on propose dans la prochaine partie une protection de symboles sources déjà quantifiés (issus de codeurs en couche existants), en utilisant la transformation Mojette.

[25] Un système PET[Albanese 96] est placé en sortie du quantificateur à description multiple.

Deuxième partie

Système de protection inégale par représentation Mojette

Introduction de la deuxième partie

Dans le cadre de la transmission ou du stockage d'une source hiérarchique multi composante, la conduite à tenir est *a priori* de protéger chaque composante en fonction de son importance. C'est du reste ce que l'on fait naturellement lors d'une expression orale où la phrase clé est appuyée par une pause, une élévation de voix ou une transition qui attire l'attention. En termes de transmission numérique, l'ajout d'information relative à la robustesse des composantes d'une source peut être effectué à l'aide de codes à protection inégale. La littérature fait largement référence à ce mécanisme de protection en justifiant son utilité par la sémantique différenciée des informations (au niveau région) des codeurs source de deuxième génération et par les propriétés les canaux de communication actuelles (communication mobile, communication multicast). Les méthodes de protection contre les erreurs de transmission par codes détecteur ou correcteur d'erreurs binaires peuvent être bien sûr utilisées. Cependant, on s'aperçoit vite de leur limite avec la transmission par paquets sur un réseau qui induit des pertes de paquets d'information et non plus des erreurs isolées.

L'apport des codes MDS est manifeste dans ce contexte de par la solution optimale qu'ils fournissent. Les paquets de message et de redondance sont dès lors indiscernable au vu du canal et présente les mêmes pouvoirs de reconstruction à la réception. Néanmoins, l'arithmétique des corps de Galois qui construit cette famille de codes peut s'avérer restrictive dans certain cas. Tout les couples (m, n) où m paquets parmi n reconstruisent la composante ne sont en effet pas permis. L'optimisation du codage canal pour une granularité de source donnée est dans ce cas limitée.

La transformation Mojette introduite au chapitre précédent ne considère pas un corps d'éléments finis particuliers. On parlera de ce fait d'ixel pour suggérer un élément contenu sur un support d'informations sans hypothèse sur le nombre de bits qui le définit. Il est de plus possible à l'opérateur de générer une projection supplémentaire par rapport à un sous ensemble donné. Dès lors, tout couple (m, n) où m et n sont des entiers peuvent être utilisés dans la protection d'une composante de la source. On utilisera la notation $Moj(m, n)$ pour faire référence au code dont un nombre m de projections suffit pour la reconstruction parmi n envoyées. La redondance introduite sera fonction de l'importance de la composante et des caractéristiques du canal.

On se propose dans le premier chapitre de cette deuxième partie de construire au sens

géométrique du terme le code $Moj(m, n)$. L'étape d'allocation des ixels sur un espace 2D correspond à la mise en mémoire tampon des éléments sources avant leur transmission. Une question se pose sur la forme de la mémoire dans cet espace 2D en fonction de la protection demandée. La particularité de cette construction est de générer des projections indiscernables au vu du canal. De ce fait, cette méthode de protection égale est parfaitement adaptée à la transmission sur des canaux à qualité de service non garantie. Une première comparaison, avec les codes MDS, est réalisée à l'issue de ce chapitre 4.
La protection simultanée de composantes hiérarchisées est mise en évidence par de nouvelles mémoires tampon "géométriques" qui supportent une reconstruction graduable à la réception. La composante prioritaire sera reconstruite en premier avec un nombre de projections relativement faible et la composante dite de basse priorité nécessitera un plus grand nombre de projections pour effectuer l'inversion. La conception des supports d'information sera en outre contrainte par la taille des projections. Le raffinement qu'apporte chaque projection dans la reconstitution de l'information originale témoigne des propriétés de description multiple que possèdent cette forme de protection graduable.
Dans un dernier chapitre, nous apporterons une optimisation dans l'affectation du nombre de projection suffisantes pour la reconstruction de chaque composante. Une comparaison avec le schéma de protection égale valide l'intérêt de la description multiple de l'information. La construction des mémoires tampon géométriques à reconstruction graduable et l'optimisation proposée pose les briques fondamentales de notre système de protection inégale par transformation Mojette.

Chapitre 4

Protection Mojette non graduable

Sommaire

4.1 Choix du support et choix des projections 78

4.1.1 Protection Mojette 78

4.1.2 Propositions de mémoire tampon géométrique 79

4.2 Représentation Mojette et code MDS 91

4.2.1 Expression du coût supplémentaire par rapport aux codes MDS 91

4.2.2 Résultats 93

4.3 Conclusion 96

La mise en place d'un réseau sécurisé nécessite une gestion de l'intégrité des informations tant dans sa composante stockage que dans sa composante transmission. Cette intégrité peut être à la fois menacée par des agressions volontaires ou fortuites. Notre objectif ici est de préserver l'information dans le réseau lui-même. La stratégie habituelle dans ce cas est de répéter l'information pour assurer sa transmission avec un critère minimum de fidélité. En dessous de ce critère, le récepteur peut demander la retransmission de segments d'information égarés. Cependant, les contraintes de disponibilité inhérentes aux applications temps réel interdisent cette répétition temporelle. On préferera sous ces contraintes une duplication spatiale qui consiste à produire un volume redondant dès l'émission. De ce fait, la source augmente certes son débit d'émission par l'introduction d'éléments correcteurs mais s'avère plus robuste face à l'effacement de données.
Dans ce chapitre, la protection est faite sur le volume global de données à transmettre. Elle est évaluée par une capacité de reconstruction pour un taux de redondance et un taux de pertes données. On affinera cette protection au chapitre 5 en affectant un taux de redondance à chaque composante du volume initial en fonction de son importance (protection graduable). Comme nous l'avons vu au chapitre précédent, la transformation Mojette réalise une transformation redondante de l'information adaptée à la transmission sur réseaux de paquets. En supposant les pertes d'une projection équiprobables, les projections générées sont envisagées équivalentes en termes de reconstruction. Au terme du codage,

le volume d'information transféré sur un support 2D sera donc reconstructible à partir d'un nombre fixé de projections caractérisant le niveau de protection. Le nombre total de projections transmises sera fonction des propriétés dynamiques du canal (pertes, gigues, délais).

Le caractère redondant que revêt la transformation Mojette est un dénominateur commun avec les codes correcteurs classiques (de Hamming à Reed-Solomon). Après avoir satisfait les fonctionnalités décrites ci-dessus par un choix idoine du support et des projections, on s'attardera sur la comparaison du code Mojette présenté avec le code optimal MDS pour dégager la propriété $(1+\varepsilon)$MDS que possède la transformation Mojette.

4.1 Choix du support et choix des projections

4.1.1 Protection Mojette

Comme nous l'avons vu au chapitre 3, la transformation Mojette présente d'excellentes propriétés lorsqu'elle est utilisée pour la transmission de données sur des réseaux de paquets (complexité opératoire, redondance, génération d'entités indépendantes...). L'utilisation d'origine de l'opérateur nous permet de générer des projections à partir d'un support rectangulaire. Une fois calculée, chaque projection est transmissible comme unité de transport sur un réseaux de paquets. Chaque élément du support contient, de manière indifférente, un ixel corrélé ou non corrélé avec ses voisins. Dans le premier cas, la structure des données peut être utilisée pour définir la forme et la capacité du support. Une image offre par exemple, par ses dimensions, un support naturel projetable. Pour transmettre une image rectangulaire, on projette ainsi directement la forme et son contenu dans les unités de transport. Dans le cas de conditions de reconstruction non respectées, cette solution présente l'avantage de retrouver une version dégradée de l'image par l'utilisation de la corrélation conservée dans les projections [Philippé 98]. Pour transmettre un signal 1-D, la corrélation des échantillons dirige l'allocation sur un support rectangulaire. Dans le deuxième cas d'ixels non corrélées, le volume d'information à transmettre détermine simplement la capacité du support. Quelle que soit la nature de la source, la transmission par transformation Mojette s'effectuera par les trois tâches nécessaires suivantes :

- allocation des données sur un support ;
- projection du support ;
- transmission des projections.

Dans la première étape d'allocation, le support se comporte comme une mémoire tampon. La position dans un espace à deux dimensions où sont alloués les ixels détermine le niveau de protection (le nombre de projections nécessaires). On parle ainsi de mémoire tampon géométrique.
La description obtenue dans la seconde étape, permet une protection grossière de la source

au vu des exemples donnés ci-dessus avec un support rectangulaire. Un ensemble suffisant de projections peut être en effet déterminé par les conditions de reconstruction (critère de Katz 55) et une projection supplémentaire est calculable (corollaire du théorème Normand-Guédon **??**). Le nouvel ensemble obtenu forme une description redondante de la source qui autorise la perte d'une projection au cours de la transmission.
On se propose de préciser ce résultat pour un ensemble de N projections tel qu'un sous-ensemble de M projections reconstruisent la mémoire tampon. La redondance se traduira par $N - M$ projections supplémentaires équivalentes. Chaque élément de la mémoire nécessite au plus M projections pour être reconstruit avec les C_M^N cas possibles de recevoir M projections parmi N. De cette manière, les N descriptions générées sont équivalentes en termes de reconstruction. On définit l'indice de protection comme étant la proportion du nombre de projections nécessaires sur le nombre de projections envoyées soit $\frac{M}{N}$.

4.1.2 Propositions de mémoire tampon géométrique

Support rectangulaire

Sans restriction, on choisit de satisfaire la contrainte verticale du critère de Katz à savoir $Q \leq \sum_{i=1}^{N} |q_i|$. Les mêmes résultats pourraient être obtenus en respectant la contrainte horizontale. Si un nombre M de projections suffit à reconstruire le support rectangulaire, la contrainte verticale peut être restreinte à l'égalité soit $Q = \sum_{i=1}^{M} |q_i|$. L'équivalence de toutes les combinaisons de M projections est donc assurée par des q_i constants pour tout $i = 1, 2, ...N$. Supposons en effet pour un rectangle avec $Q = 4$ (figure 4.1), que l'on ait dans des sous-ensembles reconstructibles de projections les valeurs $q_i = \{1, 2, 1\}$ (sous-ensemble (2) et (3) figure 4.1) parmi un ensemble S de $N = 5$ projections :

$$S = \{(-2, 1), (-1, 1), (0, 1), (1, 2), (2, 1)\}.$$

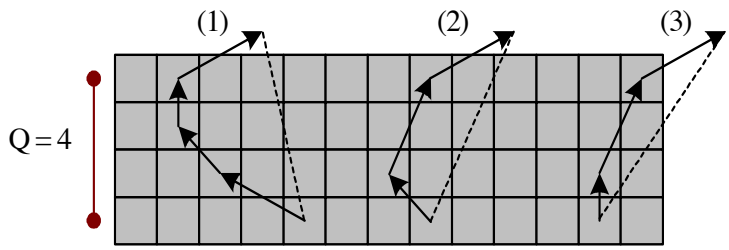


Fig. 4.1: Représentation des vecteurs de projections (résultante en pointillé) sur un support rectangulaire avec $Q = 4$ pour trois sous-ensembles reconstructibles parmi l'ensemble $S = \{(-2, 1), (-1, 1), (0, 1), (1, 2), (2, 1)\}$. (1) : $\{(-2, 1), (-1, 1), (0, 1), (2, 1)\}$; (2) : $\{(-1, 1), (0, 1), (1, 2)\}$; (3) : $\{(0, 1), (1, 2), (2, 1)\}$.

Les valeurs p_i n'ont ici pas d'influence dans la reconstruction. Pour ces combinaisons, seulement trois projections sont nécessaires au lieu de quatre pour l'autre combinaison (sous-ensemble (1) figure 4.1) qui ne contient pas l'angle où $q = 2$.

Aussi pour satisfaire à la fois les conditions d'équivalence et de reconstruction, on choisit q_i constant. Les angles de projections sont donc du type $S = \{(p_i, q)\ \forall i = 1, 2, ...N\}$. Des supports rectangulaires avec une hauteur Q de Mq ixels sont reconstructibles par tout les sous-ensembles de S de M projections. Initialement la valeur q a été choisie sous réserve que $PGCD(p_i, q) = 1\ \forall i = 1, 2, ...N$. En outre, de par la périodicité, les projections (p, q) et $(-p, -q)$ sont identiques. On a donc réduit le choix des projections à $q > 0$. La taille d'une projection pour un support rectangulaire de dimension $P \times Q$ est donnée par :

$$\#bins_i = |p_i|\,(Q-1) + q(P-1) + 1. \tag{4.1}$$

Le nombre total de bins est simplement la somme des bins contenus dans toutes les projections, soit :

$$\#bins = \sum_{i=1}^{N} bins.$$

Ce nombre total de bins sera d'autant plus faible à q constant et pour des dimensions de support rectangulaire donné que les valeurs de p_i seront faibles (éq. 4.1). Autrement dit, la contrainte horizontale est fortement relachée pour minimiser la redondance. Aussi, on a :

$$\sum_{i=1}^{N} |p_i| \ll P \tag{4.2}$$

On choisit pour ce faire des valeurs de p consécutives sur Z et centrées en 0. On distingue deux cas en fonction de la parité de N :

$$\begin{aligned} p_i &\in [-\frac{N-1}{2}, \frac{N-1}{2}]\ \forall i = 1, 2...N \text{ avec } N \text{ impair}, \\ p_i &\in [-\frac{N}{2}+1, \frac{N}{2}]\ \forall i = 1, 2...N \text{ avec } N \text{ pair}. \end{aligned}$$

De même, on distingue deux sommes A et B des valeurs absolues de p_i :

$$\begin{aligned} A &= \sum_{i=1}^{N} |p_i| \text{ avec } N \text{ impair}, \\ &= (\frac{N-1}{2})(\frac{N+1}{2}), \\ &= \frac{N^2-1}{4} \text{ avec } N \text{ impair}. \end{aligned} \tag{4.3}$$

$$\begin{aligned} B &= \sum_{i=1}^{N} |p_i| \text{ avec } N \text{ pair,} \\ &= \frac{N}{2}(\frac{N}{2}+1) - \frac{N}{2}, \\ &= \frac{N^2}{4} \text{ avec } N \text{ pair.} \end{aligned} \tag{4.4}$$

En termes de capacité, le nombre d'ixels contenu dans la mémoire tampon est de $Q \times P$ tel que :

$$\#ixels = Mq \times P.$$

Aussi, l'incrémentation de la dimension P entraîne une augmentation de q bins dans les N projections alors que la capacité augmente de Mq ixels. La redondance s'exprime par le rapport de la somme du nombre total de bins contenus dans toutes les projections et de la capacité de la mémoire soit :

$$\begin{aligned} Red &= \frac{\#bins}{\#ixels} - 1 \\ Red &= \frac{\sum_{i=1}^{N} (|p_i|\,(Q-1) + q(P-1) + 1)}{Q \times P} - 1 \\ &= \frac{N(q(P-1)+1) + (Mq-1)\sum_{i=1}^{N} |p_i|}{qMP} - 1 \end{aligned} \tag{4.5}$$

Sur la figure 4.2 est représenté le comportement de Red pour $q = \{1, 5, 11, 27\}$ en fonction de la capacité d'une mémoire tampon géométrique rectangulaire (en nombre d'ixels) limité à 5 Ki (5120 ixels). La mémoire tampon est reconstructible ici par 4 projections parmi 7 (Moj(7,4)) prises dans la série

$$S = \{(-4,q), (-2,q), (-1,q), (0,q), (1,q), (2,q), (4,q)\} \text{ soit } M = 4 \text{ et } N = 7.$$

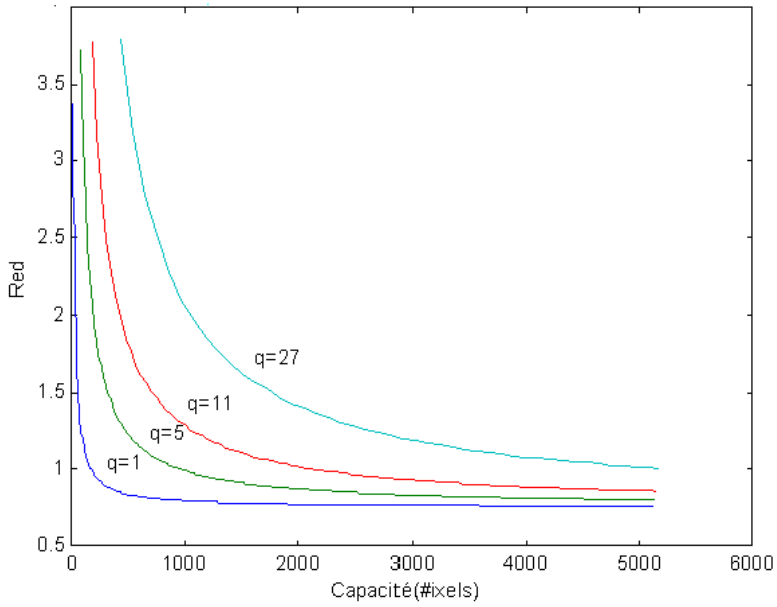


Fig. 4.2: Comportement de la redondance pour q = {1, 5, 11, 27} en fonction de la capacité d'une mémoire tampon géométrique rectangulaire.

On peut remarquer sur la figure 4.2 que toutes les courbes ont une même asymptote horizontale. Ce comportement se justifie par la limite de *Red* lorsque la capacité est infinie *i.e* P infini. En effet, on a :

$$\lim_{P \to +\infty} Red = \frac{N}{M} - 1 \tag{4.6}$$

Dans l'exemple de la figure 4.2, on a bien l'asymptote en $\frac{7}{4} - 1$ soit à 75 % de redondance qui supporte le manque d'au plus trois projections parmi sept soit 43 % de manque de projections. À titre de comparaison, le Moj(5,4) abordé au début de paragraphe présente une asymptote à 25 % et autorise la perte de 20 % de projections. Le ratio de la redondance sur le taux de perte est loin d'être constant. Une augmentation linéaire du taux de pertes nécessite pour chaque incrément une plus forte augmentation de la redondance.
Cette asymptote est atteinte plus rapidement lorsque $q = 1$. La redondance est donc minimale lorsque $q = 1$, pour un niveau de protection donné et pour un volume d'information à protéger de même taille. Du point de vue géométrie discrète, prendre $q > 1$ revient à sommer des ixels placés sur des droites non connexes. Chaque sommation correspond en fait à des parcours sur des supports distincts (figure 4.3). Dans la suite, on choisira $q = 1$ pour tous les angles de la série[26]. La mise en œuvre de la transformation Mojette directe détaillée dans l'annexe 1 page **??** sera en outre facilitée. La redondance de l'équation 4.5

[26] Cependant, on n'interdira pas l'utilisation d'autres valeurs de q dans un comportement asymptotique *i.e.* pour des mémoires de très grande capacité.

dans ce cas se réduit à :

$$Red = \frac{NP + (M-1)\sum_{i=1}^{N} |p_i|}{MP} - 1 \tag{4.7}$$

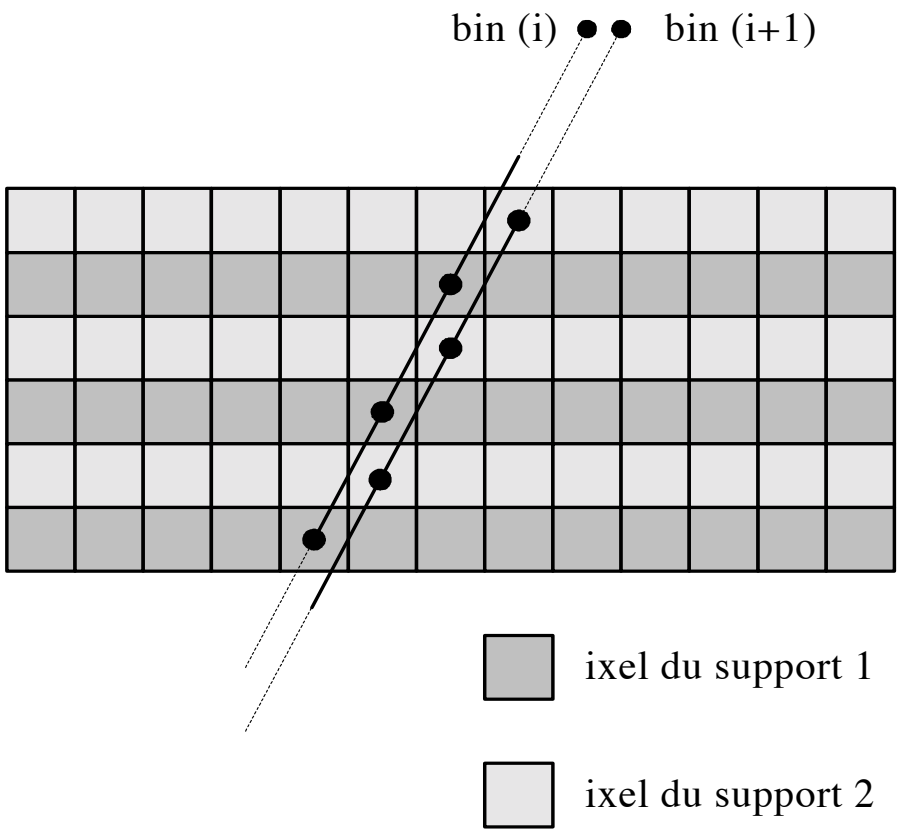


Fig. 4.3: Représentation des parcours sur deux supports distincts pour la projection (1,2).

– *exemple d'un support 6×3 reconstructible par trois projections parmi cinq*

L'exemple de la figure 4.4 illustre la protection d'un vecteur source $\{X_j\}_{j=1}^{15}$ avec Moj(5,3). La figure 4.4(a) représente l'allocation du vecteur source dans la mémoire tampon géométrique. Le critère de Katz est satisfait pour sa contrainte verticale. En effet, on a

$$\sum_{i=1}^{M} |q_i| = 3 = Q.$$

Le support est donc reconstructible avec trois projections parmi cinq. On dénombre C_3^5 soit dix réalisations de trois projections parmi cinq. Pour l'ensemble des réalisations, chaque position dans la mémoire tampon est étiquetée par M, le nombre minimal de projections

nécessaires à la reconstruction dans la figure 4.4(b). Ce nombre pour un support rectangulaire est constant quelle que soit la position dans la mémoire.

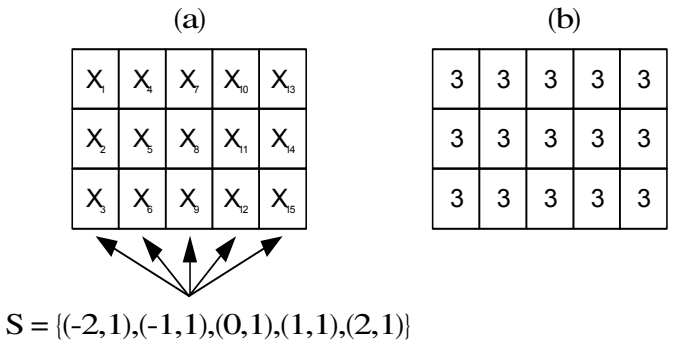


Fig. 4.4: (a) Exemple d'une mémoire tampon géométrique rectangulaire 6x3. (b) chaque élément de la mémoire est exactement reconstructible par trois projections prises dans l'ensemble S.

Cette première mémoire tampon géométrique présente l'avantage d'avoir une forme facilement décrite par sa longueur et sa hauteur. L'allocation du vecteur source se réduit ainsi à une copie d'éléments dans un tableau 2-D. Cependant, les descriptions équivalentes générées ne sont pas de tailles constantes. On a ainsi dans l'exemple ci-dessus des tailles de 9, 7, 5, 7 et 9 bins pour les cinq projections de S. La proposition ci-dessous offre des descriptions de tailles constantes sans trop ajouter de complexité à la forme de la mémoire tampon.

Support hexagonal

Dans le cas d'un support autre que rectangulaire, le nombre de bins est calculé par la différence des abscisses du premier et du dernier bin plus un. Ainsi pour une projection donnée, si $m_1$et m_2 représentent respectivement les abscisses du premier et du dernier bin, on a :

$$\#bins = |m_2 - m_1| + 1. \tag{4.8}$$

Or l'équation de la droite par lequel passe tout les pixels projetés est $m = -qk + pl$.
Si quelles que soient les N projections de la série, deux ixels de coordonnées (k_1, l_1) et (k_I, l_I) se projettent toujours dans le premier et dernier bin, ce nombre de bins devient :

$$\begin{aligned} \#bins_i &= |-q_i k_I + p_i l_I - (q_i k_1 + p_i l_1)| + 1 \\ &= |-q_i \Delta k + p_i \Delta l| + 1 \ \forall i = 1, 2, ...N. \end{aligned} \tag{4.9}$$

En outre, on convient que $\Delta l = 0$ si ces deux ixels sont choisis sur la même ligne de la mémoire tampon et donc que le nombre de bins explicité à l'équation 4.9 devient :

$$\begin{aligned} \#bins_i &= |-q_i \Delta k| + 1 \\ &= |q_i| \, \Delta k + 1 \ \forall i = 1, 2, ...N. \end{aligned} \tag{4.10}$$

Le nombre de bins donné à l'équation 4.10 est valable dès lors que :

- l'allocation d'un vecteur source $\{X_j\}_{j=1}^{I}$amène les ixels X_1 et X_I à être projetés dans toutes les projections d'une série S respectivement dans le premier et dans le dernier bin ;
- les coordonnées de $X_1 \left| \begin{matrix} k_1 \\ l_1 \end{matrix} \right.$ et $X_I \left| \begin{matrix} k_I \\ l_I \end{matrix} \right.$ dans le repère $(\mathrm{O}, \overrightarrow{k}, \overrightarrow{l})$ sont telles que $l_1 = l_I$.

L'équivalence de tous les sous-ensembles d'exactement M projections en termes de reconstruction implique le respect des conditions données par la morphologie mathématique pour chaque sous-ensemble (chapitre 3). Chaque dilaté correspondant à un sous-ensemble ne doit pas être inclu dans le support hexagonal considéré. Le nombre minimal M de projections induit une hauteur constante pour chaque dilaté égale à la hauteur du support plus un élément. La valeur des q pour chaque angle est donc constante. La taille des projections devient alors avec $q > 0$:

$$\#bins = q\Delta k + 1. \tag{4.11}$$

Avec les mêmes considérations que pour le support rectangulaire, on fixe $q = 1$. Les parcours des vecteurs de projection sur le support s'effectuent ainsi en sommant un ixel de chaque ligne. Ce choix permet d'obtenir les projections de plus petite taille ($\#bins = \Delta k + 1$). Il est aussi justifié par le taux de redondance avec néanmoins un nombre d'ixels dans le support légèrement inférieur par rapport au cas rectangulaire (figure 4.5(a)).
La contrainte sur la projection du premier et du dernier ixels ($\{X_1\}$ et $\{X_I\}$) borne les valeurs consécutives de p sur $[p_{\mathsf{min}}, p_{\mathsf{max}}]$ telles que $\{X_1\}$ et $\{X_I\}$ soient contenus dans les premier et dernier bins de chaque projection. Aussi, pour l'ensemble S de projections, on a :

$$S = \{(p_{\mathsf{max}}, 1), ((p_{\mathsf{max}} - 1, 1)...((p_{\mathsf{max}} - N + 1, 1),$$

Pour des valeurs de p_{min} dépendantes de la parité de N

$$\begin{aligned} p_{\mathsf{min}} &= -p_{\mathsf{max}} \text{ si } N \text{ est impair,} \\ p_{\mathsf{min}} &= -p_{\mathsf{max}} + 1 \text{ si } N \text{ est pair.} \end{aligned}$$

– *exemple d'un support hexagonal reconstructible par trois projections parmi cinq*

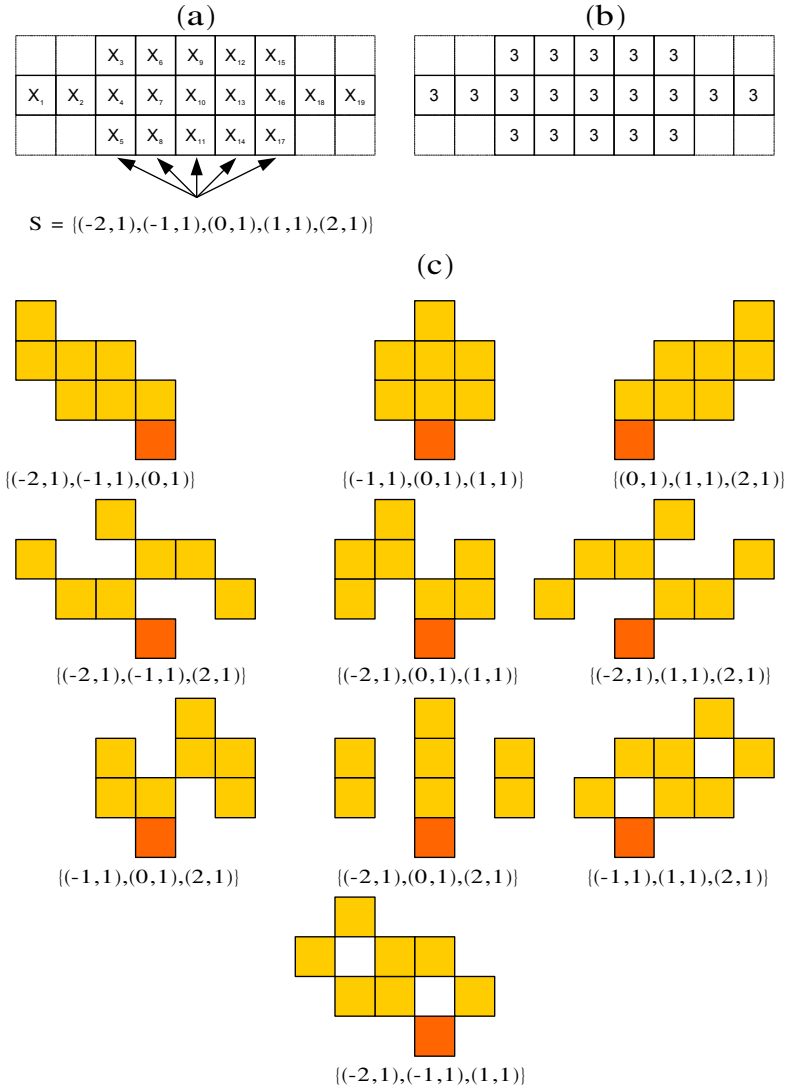


Fig. 4.5: (a) Exemple d'une mémoire tampon géométrique hexagonal. (b) Chaque élément de la mémoire est reconstructible par trois projections prises dans l'ensemble S. (c) les 10 dilatations des ES2P correspondant aux 10 combinaisons de recevoir 3 projections parmi 6.

La figure 4.5 présente l'allocation d'un vecteur source $\{X_j\}_{j=1}^{19}$ sur un convexe à 2 dimensions (a) et ses niveaux de protections (en nombre de projections nécessaires) (b). La série des projections est identique au cas précédent. L'ensemble des dix sous-ensembles possibles de projections est représenté en (c) par les dilatés des ÉS2P. Chaque sous-ensemble est suffisant pour la reconstruction. Chaque dilaté de la figure 4.5(c) possède de manière suffisante pour la reconstruction une hauteur de quatre pixels. Une valeur de $q = 2$ pour un seul angle (p, q) de la série entraine des sous-ensembles de projections reconstructibles

dans lequel seulement deux projections sont nécessaires sur les trois. La hauteur de chaque dilaté est constante. Cette hauteur excède de seulement un pixel l'épaisseur du support dans le cas d'un nombre suffisant de projections. Dans l'exemple, les ixels X_1 positionné en (k_1, l_1) et X_{19} positionné en (k_{19}, l_{19}) sont toujours placés aux extrémités de toutes les projections de l'ensemble S.

Cette deuxième mémoire tampon avec un niveau de protection présente l'avantage de produire en le projetant des descriptions de taille constante (éq. 4.11). Cette propriété est souvent requise par les protocoles de réseaux de paquets (ATM, protocole ionosphérique). Pour ce faire, la forme (et donc l'allocation) est légèrement plus complexe et le taux de redondance se trouve faiblement augmenté. La mémoire tampon géométrique est en effet plus petite en capacité. Sur la figure 4.5(a), la mémoire est occupée par 19 éléments alors que l'équivalent rectangulaire aurait pu contenir 27 éléments. Néanmoins, les rapports de capacité s'atténuent fortement avec un volume source important.

Construction de la mémoire tampon géométrique hexagonale

– *Forme*

On dispose de N projections pour réaliser le codage. Le niveau de protection impose une reconstruction par M projections parmi N où $M \leq N$. Le support aura donc une épaisseur de M ixels comme indiqué sur la figure 4.6.

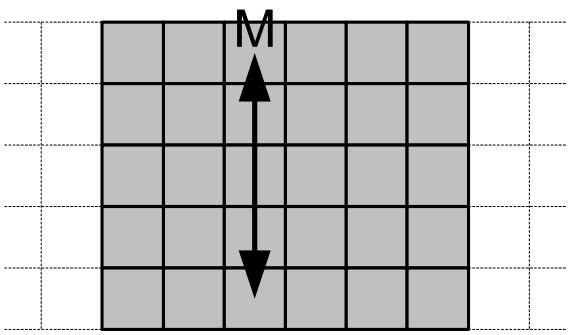


Fig. 4.6: La reconstruction du support par M projections implique une épaisseur de M ixels.

Les parties convexes (de gauche et de droite) du support sont déterminées par les deux angles (p, q) extrêmes de la série : le plus petit et le plus grand par rapport à l'axe (O, $\overrightarrow{p}$) soient (p_{min}, q) et (p_{max}, q). En d'autres termes, les droites d'équation $m = -qk + p_{\mathsf{min}}l$ et $m = -qk + p_{\mathsf{max}}l$ délimitent le support pour sa partie convexe comme représenté sur la figure 4.7. Sur cette figure, les indices $m_{1,1}$ et $m_{1,2}$ sont respectivement l'indice du premier bin de la projection (p_{min}, q) numéroté 1 et l'indice du premier bin de la projection (p_{max}, q)

numéroté 2. Si $X_1 \left| \begin{matrix} k_1 \\ l_1 \end{matrix} \right.$ on a :

$$\begin{aligned} m_{1,1} &= -qk_1 + p_{\mathsf{min}} l_1 \\ m_{1,2} &= -qk_1 + p_{\mathsf{max}} l_1. \end{aligned}$$

Et un point $X_j \left| \begin{matrix} k \\ l \end{matrix} \right.$ appartient à la partie convexe du support si :

$$\begin{cases} m_{1,1} \leq -qk + p_{\mathsf{min}} l \\ \qquad et \\ m_{1,2} \leq -qk + p_{\mathsf{max}} l \end{cases} \tag{4.12}$$

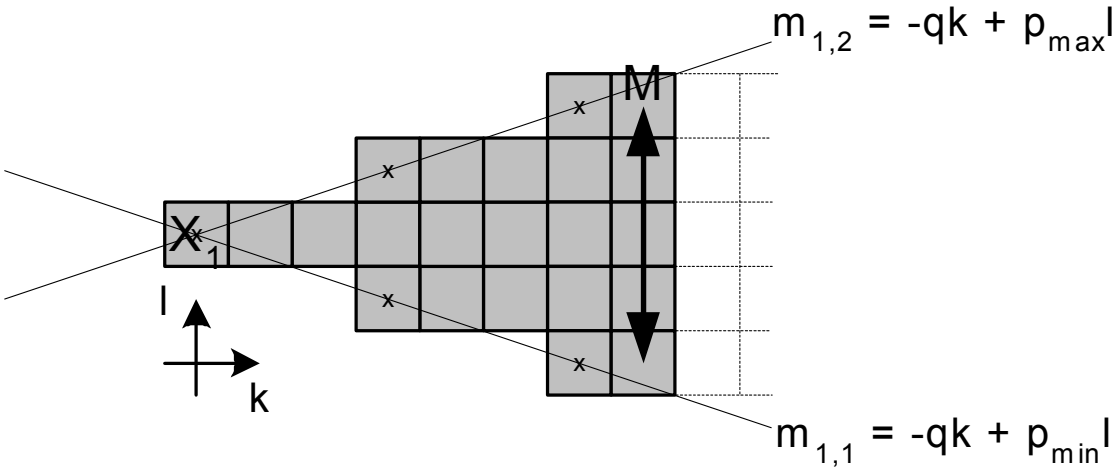


Fig. 4.7: Construction de l'extremité gauche du support hexagonal. L'ixel X1 est dans le premier bin de toutes les projections.

La partie droite du support est construite par des inégalités similaires à l'équation (4.12). Ainsi en fin de construction, on obtient un support dont la forme est représentée figure 4.8.

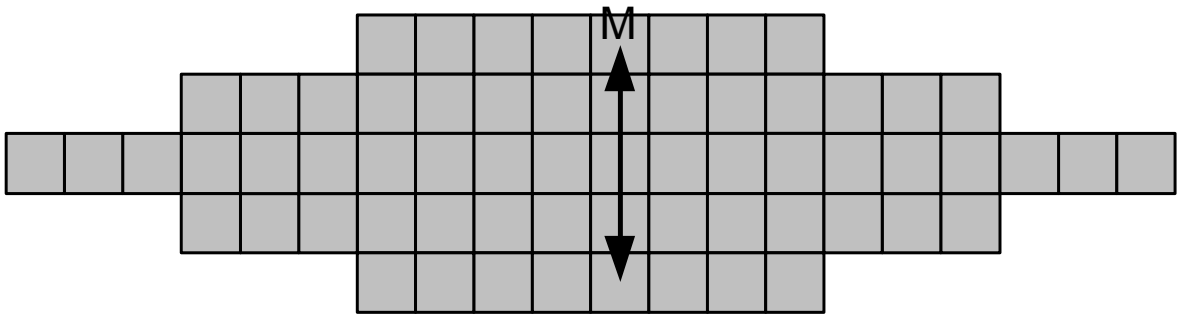


Fig. 4.8: Forme de la mémoire tampon géométrique hexagonal.

En particulier, le support de la figure 4.8 est construit à partir de la série

$$S = \{(-3,1)\ (-2,1)\ (-1,1)\ (0,1)\ (1,1)\ (2,1)\ (3,1)\}.$$

Il est reconstruit à partir de cinq projections parmi sept de S. Il permet la protection d'un vecteur $\{X_j\}_{j=1}^{64}$ soit l'allocation de 64 ixels. Les projections ont une taille de 20 bins chacune. Le coût de la protection est élevé dans cet exemple puisque le taux de redondance pour un sous-ensemble recontructible s'élève à $\frac{20\times5}{64} - 1$ soit 0,5625. Ce taux se réduit pour une mémoire tampon géométrique ayant les même propriétés de protection mais de 604 ixels (et pour des projections de 128 bins) à $\frac{128\times5}{604} - 1$ soit 0,05960.

– *Capacité*

Par ces simples mesures et par la définition du taux de redondance (éq. 3.7), on observe que le coût de la protection diminue lorsque la capacité du support augmente. Cette capacité est étudiée dans ce qui suit.

Le théorème de Pick stipule que le nombre de points discrets contenus dans un polygone A non vide et sans trou est donné par [Wells 96] :

$$Card(A) = S(A) + \frac{Card(\partial A)}{2} + 1 \tag{4.13}$$

Dans cette expression, $S(A)$ représente l'aire de l'enveloppe convexe de A, ∂A est le bord de A. Pour le polytope représenté sur la figure 4.9(a), le nombre total de points est 64, l'aire du polytope (décomposé en 3 quadrilatère figure 4.9(b)) est 52 et on dénombre 22 points de bords.

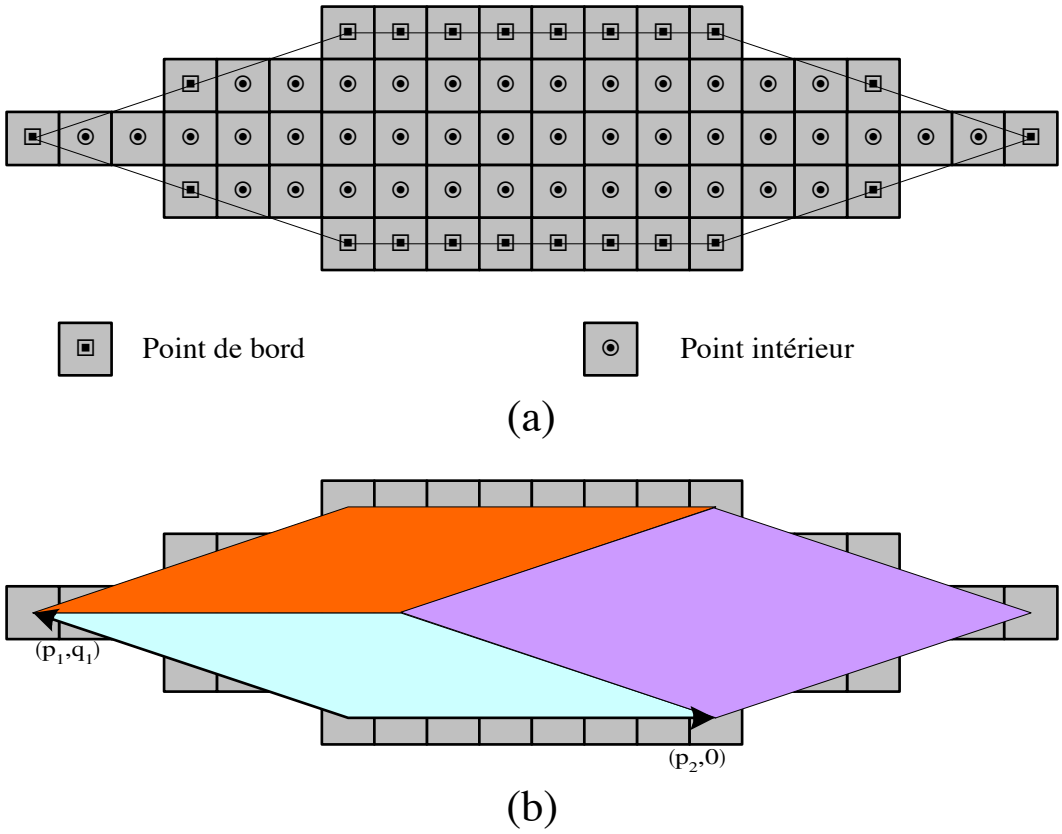


Fig. 4.9: Points intérieurs et calcul de surface d'un polytope.

La surface du polytope peut s'exprimer comme la somme des surfaces des trois paralléllogrammes comme indiqué sur la figure 4.9.

Ainsi,

$$\begin{aligned} S(A) &= 2\left|\det\begin{vmatrix} p_1 & q_1 \\ p_2 & 0 \end{vmatrix}\right| + \left|\det\begin{vmatrix} p_1 & q_1 \\ -p_1 & q_1 \end{vmatrix}\right| \\ S(A) &= 2(|p_2 q_1| + |p_1 q_1|). \end{aligned}$$

De ce calcul de surface, nous pouvons déduire de l'équation 4.13 la capacité du support hexagonal. Cette relation entre espace continu et espace discret permet de quantifier la redondance introduite par la transformation Mojette de la mémoire tampon géométrique et d'évaluer le sur-débit par rapport à des codes optimaux décrit dans la théorie des codes correcteurs.

4.2 Représentation Mojette et code MDS

4.2.1 Expression du coût supplémentaire par rapport aux codes MDS

À première vue, la Mojette est MDS dans le sens ou elle nécessite le même nombre de bins que d'ixels pour être inversible. Mais cette affirmation sous-entend que l'on ne parle plus de projections mais d'ensemble de bins (réservoir de bins). Et dans ce cas, les critères de reconstruction ne sont plus applicables. Dans la transformation Mojette classique, on ne peut en effet assurer l'inversion en prenant m bins quelconques dans un réservoir de n bins. Cependant, on peut observer que pour un ensemble suffisant de M projections, le rapport entre le nombre de bins et le nombre d'ixels à protéger peut être très faible. Il nous faut donc un peu plus de bins que d'ixels pour l'inversion. Ainsi, le code proposé approche la solution optimale MDS : il faut un peu plus de bins dans un ensemble suffisant que d'ixels que les projections décrivent. Soit :

$$\varepsilon = \frac{Nombre\ de\ bins\ d'un\ ensemble\ suffisant}{Nombre\ d'ixels} - 1.$$

Pour un support rectangulaire, le calcul du paramètre ε est similaire au calcul de Red (éq. 4.7) avec une sommation sur M projections formant l'ensemble suffisant. Ainsi, pour $q = 1$, on a

$$\begin{aligned} \varepsilon &= \frac{MP + (M-1)\sum_{i=1}^{M} |p_i|}{MP} - 1 \\ &= \frac{(M-1)\sum_{i=1}^{M} |p_i|}{MP}. \end{aligned}$$

On reprécise ici les valeurs p_i et les tailles de projections correspondantes pour $i = 1, 2, ...N$. Pour $q = 1$, on a

$$\begin{aligned} p_i &\in \left[-\frac{N-1}{2}, \frac{N-1}{2}\right] \text{ pour } N \text{ impair,} \\ p_i &\in \left[-\frac{N}{2}+1, \frac{N}{2}\right] \text{ pour } N \text{ pair.} \end{aligned}$$

Et pour des dimensions de rectangle $Q \times P$, la taille d'une projection est

$$\#bin_i = |p_i|\,(Q-1) + P.$$

Le nombre de bins total minimal est obtenu pour M projections parmi N avec

$$\begin{aligned} p_i &\in \left[-\frac{M-1}{2}, \frac{M-1}{2}\right] \text{ pour } M \text{ impair,} \\ p_i &\in \left[-\frac{M}{2}+1, \frac{M}{2}\right] \text{ pour } M \text{ pair.} \end{aligned}$$

Selon la parité de M et par éq.4.4 et éq.4.3, donnant pour M projections

$$\begin{aligned} \sum_{i=1}^{M} |p_i| &= \frac{M^2-1}{4} \text{ avec } N \text{ impair,} \\ \sum_{i=1}^{M} |p_i| &= \frac{M^2}{4} \text{ avec } N \text{ pair,} \end{aligned}$$

on obtient deux valeurs minimales pour ε, soit $\varepsilon_{\mathsf{min}}$:

$$\varepsilon_{\mathsf{min}} = \frac{(M-1)(M^2-1)}{4MP} \text{ pour } M \text{ impair,} \tag{4.14}$$

$$\varepsilon_{\mathsf{min}} = \frac{(M-1)M}{4P} \text{ pour } M \text{ pair.} \tag{4.15}$$

La valeur maximale pour ε est donnée dans le cas de la réception des M projections de plus grande taille soit pour M valeurs de p_i les plus élevées parmi les N envoyées. Le calcul de cette borne supérieure n'est pas détaillée ici. Des valeurs sont cependant indiquées pour la protection Moj(7,5) dans le tableau 4.2 à la page 94.

Pour un support hexagonal, le calcul du paramètre ε diffère du cas rectangulaire dans l'expression de nombre total de bins et par une capacité d'ixels légérement plus faible. Cette différence d'ixels est représentée par une texture pointillée sur la figure 4.10.

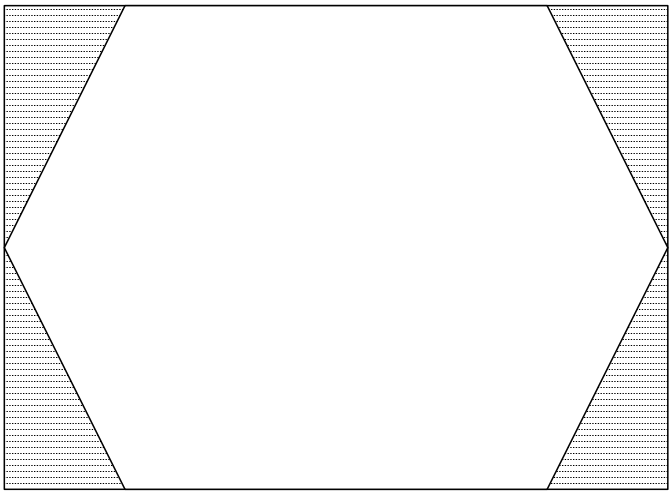


Fig. 4.10: Différence de capacité entre la mémoire tampon hexagonale et la mémoire tampon rectangulaire.

Si on pose δ, cette différence d'ixels entre le cas rectangulaire et le cas hexagonal, en prenant la taille d'une projection obtenue à l'éq.4.11 avec $P = \Delta k + 1$, on a :

$$\varepsilon = \frac{M(q(P-1)+1)}{qMP - \delta} - 1. \tag{4.16}$$

À $q = 1$, cette expression du paramètre ε se réduit à :

$$\begin{aligned} \varepsilon &= \frac{MP}{MP - \delta} - 1, \\ &= \frac{\delta}{MP - \delta}. \end{aligned}$$

Quelle que soit la dimension P, la différence d'ixels δ entre les cas rectangulaire et hexagonal est constante. Aussi, pour de grandes valeurs de P, on a

$$\lim_{P \to +\infty} \varepsilon = 0. \tag{4.17}$$

4.2.2 Résultats

Comparaison des mémoires tampon rectangulaires et hexagonales

Le tableau 4.1 présente le comportement presque MDS pour quelques capacités d'une mémoire tampon hexagonale. La source est ici un fichier à protéger de taille variable où chaque ixel est un octet. La tranformation Mojette s'effectue ainsi modulo 256. Le niveau de protection est de 5 projections nécessaires parmi 7 (Moj(7,5)). Le codage supporte ainsi la perte de 2 projections parmi 7 soit un taux d'environ 30%. La mémoire tampon de 64 ixels est identique à la représentation de la figure 4.8. La différence de capacité entre les cas hexagonal et rectangulaire est $\delta = 36$. Le tableau 4.1 nous indique la taille de la charge utile L d'un paquet (la taille d'une projection), le paramètre ε et le taux de redondance globale avec 7 projections envoyées sachant que 5 sont suffisantes (soit $N = M + 2$). L'ensemble des projections considéré est :

$$S = \{(-3,1)(-2,1)(-1,1)(0,1)(1,1)(2,1)(3,1)\}.$$

Pour le même ensemble S et pour un niveau de protection identique Moj(7,5), le tableau 4.2 présente le comportement presque MDS pour une mémoire tampon rectangulaire. Pour ce support, les projections ne sont plus de taille constante. Les tailles sont indiquées entre accolades de la projection (-3,1) à la projection (3,1). Par les différentes tailles de projections, ε varie en fonction des combinaisons de projections utilisées pour la reconstruction. Cette variation n'est pas exprimée dans l'équation (4.14) car elle suppose un nombre total de M projections. Aussi, les deux valeurs minimales et maximales pour ε respectivement

Capacité (octets)	L (#bins (o))	ε	*Red* (M|M+2)
64	20	.5625	1.1975
604	128	.0596	0.4834
1 Ko	212	.0352	0.4492
2 Ko	417	.0181	0.4253
5 Ko	1032	.0078	0.4109
10 Ko	2056	.0039	0.4055

Tab. 4.1: Mesure du paramètre ε en fonction de la capacité de la mémoire tampon hexagonale

Capacité (octets)	L (#bins (o))	ε $\{\varepsilon_{\mathsf{min}}; \varepsilon_{moy}; \varepsilon_{\mathsf{max}}\}$	Red (M|M+2)
65	{25,21,17,13,17,21,25}	{0,3692 ; 0,4308 ; 0,6769}	1,1385
605	{133,129,125,121,125,129,133}	{0,0396 ; 0,0463 ; 0,0727}	0,4793
1 Ko	{217,213,209,205,209,213,217}	{0,0244 ; 0,0283 ; 0,0439}	0,4482
2 Ko	{422,418,414,410,414,418,422}	{0,0127 ; 0,0146 ; 0,0224}	0,4248
5 Ko	{1036,1032,1028,1024,1028,1032,1036}	{0,0047 ; 0,0055 ; 0,0085}	0,4094
10 Ko	{2060,2056,2052,2048,2052,2056,2060}	{0,0023 ; 0,0027 ; 0,0042}	0,4047

Tab. 4.2: Mesure du paramètre ε en fonction de la capacité de la mémoire tampon rectangulaire

$\varepsilon_{\mathsf{min}}$ et $\varepsilon_{\mathsf{max}}$ sont représentées correspondant respectivement au sous-ensemble suffisant le plus favorable, composé des projections de plus petites tailles :

$$S^{'} = \{(-2,1)(-1,1)(0,1)(1,1)(2,1)\} \ ;$$

et au sous-ensemble suffisant le plus défavorable, composé des projections de plus grandes tailles :

$$S^{'} = \{(-3,1)(-2,1)(-1,1)(2,1)(3,1)\}.$$

La valeur moyenne du paramètre ε est aussi insérée pour toutes les combinaisons de réception *i.e.* 21 scénarios de réceptions différents.

Le paramètre ε calculé dans le tableau 4.1 est légèrement supérieur à la moyenne observée dans le tableau 4.2. La mémoire tampon hexagonale présente un surcoût par rapport à la mémoire tampon rectangulaire. Néanmoins cette légère augmentation du nombre de symboles nécessaires permet d'obtenir des projections de tailles constantes. Dans les deux cas, le paramètre ε décroît fortement avec la capacité de la mémoire. Cette observation confirme le comportement de ε en fonction de P, la dimension horizontale de la mémoire, dans l'équation (4.16) et dans les équations (4.14) et (4.15), respectivement pour

M	L (#bins (o))	ε	Red (M|M+1)	S
3	1707	8×10^{-4}	0,3344	{[-1,1][0,1][1,1]}∪{[2,1]}
4	1283	0,002	0,2529	{[-1,1][0,1][1,1][2,1]}∪{[-2,1]}
5	1029	0,004	0,2056	{[-2,1][-1,1][0,1][1,1][2,1]}∪{[3,1]}
7	742	0,014	0,1589	{[-3,1]...[0,1]...[3,1]}∪{[4,1]}
10	535	0,043	0,1483	{[-4,1]...[0,1]...[5,1]}∪{[-5,1]}
15	394	0,1529	0,2298	{[-7,1]...[0,1]...[7,1]}∪{[8,1]}

Tab. 4.3: Calcul des paramètres ε et Red pour une taille de mémoire hexagonale de 5120 éléments pour un nombre variable de projections, composé du sous-ensemble reconstrutible (calcul ε) et d'une projection supplémentaire (calcul de Red).

la mémoire hexagonale et la mémoire rectangulaire. En outre, on notera la diminution de la redondance avec l'augmentation de la capacité des mémoires pour un niveau de protection identique (Moj(7,5)).

Coût du partage de l'information

Dans le tableau 4.3, la capacité d'une mémoire tampon hexagonale est fixée à 5 Ko. Les projections sont donc de tailles constantes. Chaque ligne du tableau indique pour un niveau M de protection donné la valeur du paramètre ε. En outre, une indication de la redondance est fournie en produisant une projection supplémentatire par rapport à l'ensemble suffisant. La colonne S précise les projections choisies pour l'ensemble suffisant en ajoutant la projection supplémentaire.

.

Dans le tableau 4.3, le nombre de descriptions de l'information augmente avec le nombre de projections générées. Le partage de l'information se traduit par des descriptions plus petites mais pour un coût plus élévé (le paramètre ε augmente avec le nombre de projections). D'autre part, la redondance diminue lorsque le niveau de protection est moins élevé. Les paramètres de la première ligne supportent en effet la perte d'une projection parmi quatre (25 % de pertes) alors que les paramètres de la dernière ligne autorisent la perte d'une projection parmi quinze (6,67 % de pertes).

Influence de la représentation des éléments d'information

Plus la capacité en nombre d'ixels est élevée, plus les performances sont optimales au sens MDS. L'épaisseur du support étant contrainte par le niveau de protection (le nombre M de projections nécessaires), c'est par la longueur du support que l'on augmente sa capacité. La dimension du support offrira une solution proche de l'optimale (cas MDS) pour une longueur importante (contrainte horizontale du critère de Katz, page 55). En d'autres termes, si l'on travaille non plus comme précédemment avec des mots de 8 bits

Capacité (bits)	L (#bins (bits))	ε	*Red* (M|M+2)
514	110	0,07	0.4981
4834	974	0,007	0.4104
8 Kb	1646	0,004	0.4062
16 Kb	3284	0,002	0.4031
40 Kb	8200	9×10^{-4}	0.4012
80 Kb	16392	4×10^{-4}	0.4006

Tab. 4.4: Mesure du paramètre ε en fonction de la capacité du mémoire hexagonale

mais au niveau binaire (chaque ixel est un bit), le support, pour accueillir le même volume d'information, est beaucoup plus long (pour 8 fois plus d'ixels). La transformation Mojette dans ce cas utilise l'addition modulo 2.
Les mêmes paramètres de codage sont utilisés dans les tableaux 4.1 et 4.4 mais, pour ce dernier, c'est une une représentation binaire des éléments d'information qui est utilisée. Le codage Moj(7,5) est donc réalisé avec l'ensemble

$$S = \{(-3,1)(-2,1)(-1,1)(0,1)(1,1)(2,1)(3,1)\}$$

pour $\delta = 36$.

Le paramètre ε est nettement plus faible que dans le tableau 4.1. La représentation binaire nécessite une capacité plus grande en nombre d'éléments et entraine le comportement asymptotique mis en évidence sur la figure 4.2.

4.3 Conclusion

Dans ce chapitre, nous présentons une méthode de protection globale de l'information. C'est par dispersion de l'information que se concrétise notre méthode. L'utilisation de la représentation Mojette corrèle l'information source et compense les pertes au cours de la transmission : dès l'émission, on calcule un volume redondant d'information. Le schéma proposé s'apparente de ce fait aux codages FEC (§ 2.1). Les performances sont comparées à la solution MDS. Au dessus d'un certain taux de pertes, le volume d'information peut être reconstruit de manière déterministe par M, un nombre fixé de projections parmi N. En dessous de ce taux, les propriétés du canal interdisent toute reconstruction, même partielle.
La propriété d'équivalence des projections suppose un choix idoine du support d'information et des angles. Les formes rectangulaire et hexagonale sont successivement décrites. Au prix d'une légère augmentation de débit, la forme hexagonale engendre des projections de taille constante. Cette propriété facilite la synchronisation des échanges de données sur des

canaux extrêmement bruités (*e.g.* protocole ionosphérique) lorsqu'elle n'est pas imposée par le protocole lui-même (*e.g.* ATM). Des unités de transport de même taille sont soumises aux même mécanismes réseaux (fragmentation, dropping) ce qui laisse à supposer qu'elles ont les mêmes probabilités d'être perdues. Cette équiprobabilité des pertes justifie l'intérêt du support hexagonal par rapport au support rectangulaire de base.

Le code Mojette est $(1+\varepsilon)$MDS. À partir d'une surface de dimension $P \times M$ on peut calculer M projections d'angles $(p_i, 1)$ pour tout $i = 1, 2, ...M$, de manière suffisante pour un nombre ε de symboles supplémentaires. La redondance est incorporée en ajoutant des projections de type $(p_i, 1)$ pour tout $i = 1, 2, ...N$. Le niveau de complexité de ce code $(1+\varepsilon)$MDS est de l'ordre de $\mathsf{O}(IN)$ où $I = P \times M$. De plus, pour comparer les valeurs ε données dans les différents tableaux(§ 4.2.2), [Byers 98] indique une valeur de 0,054 dans son exemple.

À la différence du code MDS, où tous les symboles redondants proviennent de la corrélation de M symboles sources, la transformation Mojette inverse nécessite dans son algorithme un nombre de correspondances univoques (symboles sources explicites). Pour un niveau de protection donné, (qui se traduit par une épaisseur donnée du support), ce nombre de correspondances est constant. Le sur-débit engendré devient rapidement négligeable pour un volume source important.

Chapitre 5

Protection à n niveaux de priorité

Sommaire

5.1 Codage prioritaire par transformée Mojette 100

5.2 Choix du support et choix des projections 101

5.2.1 Projections concaténées 102

5.2.2 Supports concaténés 104

5.3 Autre construction de mémoire tampon géométrique pour les supports concaténés 109

5.3.1 Optimisation de la capacité du support 109

5.3.2 Nombre d'ixels frontière 111

5.3.3 Construction finale 112

5.4 Application à la transmission d'image 116

5.5 Conclusion 117

La représentation graduable de l'information s'impose aujourd'hui pour supporter l'hétérogénéité d'un réseau interconnecté. Cette structure hiérarchique met à la disposition des différents terminaux des niveaux de qualité en correspondance avec leur capacité de traitement et leur lien avec le réseau. C'est ainsi que les normes actuelles de codage d'images fixes (progressive JPEG, JPEG2K) et de vidéo (H263+, MPEG vidéo) ordonnent leur flux source. À titre d'exemple, la norme MPEG-4 vidéo prévoit une graduabilité spatiale et temporelle pour offrir des débits de 5-64 kbit/s jusqu'à 2Mbit/s pour sa transmission sur des liens très hétérogènes (plus de 10Mbit sur des réseaux locaux, 2Mbit/s pour un lien ADSL, 56 ou 33Kbit/s pour un modem). Malgré une amélioration de la résistance des flux aux effacements de données [Moccagatta 00], les codages de source proposés ne sont pas adaptés pour la transmission sur les réseaux actuels de par leur sensibilité aux pertes de paquets.

Une approche séparée vise à optimiser la compression pour ensuite effectuer une protection globale du train binaire produit sans l'observation du contenu sémantique de la source. Cette approche est aujourd'hui fortement remise en cause par la multiplication des ap-

plications temps réel et la faible connaissance des propriétés statistiques du canal. Une approche conjointe confère au contenu sémantique de la source une protection appropriée en fonction du canal emprunté. Suivant cette approche, nous proposons dans ce chapitre une méthode de protection graduable, extension de la méthode proposée au chapitre 4, visant à affecter une redondance graduable en fonction des incréments de qualité de chaque couche d'un codeur source. L'objectif est de transmettre la hiérarchie de la source de manière transparente pour le canal. Les paquets générés ont une même valeur dans la reconstruction et peuvent ainsi transiter via des réseaux à qualité de service non garantie. Comme au chapitre précédent, la protection d'une couche se traduit par un nombre fixe de projections Mojette nécessaires à sa reconstruction. L'extension à la protection simultanée de N couches par un système de transmission graduable est présentée dans la section 5.1. On détaillera le choix du support (mémoire tampon géométrique) et le choix des projections dans la section 5.2 pour obtenir cette fonctionnalité grâce à la représentation Mojette.

5.1 Codage prioritaire par transformée Mojette

Notre objectif avec ce système est de fournir une protection idoine à chaque niveau de la représentation hiérarchique d'une source en vue d'une transmission sur un réseau IP dépourvu de classes de priorités. Comme présenté dans le chapitre 4 pour une protection non graduable, les conditions de reconstruction d'une mémoire tampon géométrique définissent les différents niveaux de protection. Cette fois, une reconstruction graduable est nécessaire pour supporter la hiérarchie.

Reconstruction graduable de la mémoire tampon géométrique

Pour être reconstruite, une couche élémentaire nécessitera un nombre de projections plus faible qu'une couche de raffinement. On entend ansi par mémoire tampon géométrique un ensemble de supports affectés à la protection de chaque couche. Les propriétés géométriques de la mémoire tampon obtenue confèrent à chaque couche de la source une protection adaptée à son incrément de qualité. La reconstruction graduable du ou des supports permet de récupérer les éléments source progressivement. Le nombre de couches définit la granularité de la source. On peut par le procédé Mojette faire coïncider la réception d'une projection avec la reconstruction d'une couche en établissant des niveaux de protection consécutifs. Sur la figure 5.1 sont représentées schématiquement les étapes de codage et de décodage propres au système de protection Mojette. Le codage s'effectue par projection d'une mémoire tampon géométrique aux propriétés de reconstruction graduable. Dans l'exemple, la mémoire dispose de trois régions reconstructibles exactement par trois, quatre et cinq projections qui correspondent chacune à une couche distincte du codeur source. L'application de la transformation Mojette sur cette mémoire génère cinq projections (notées p1, p2, p3, p4, p5) de capacité de reconstruction équivalente qui

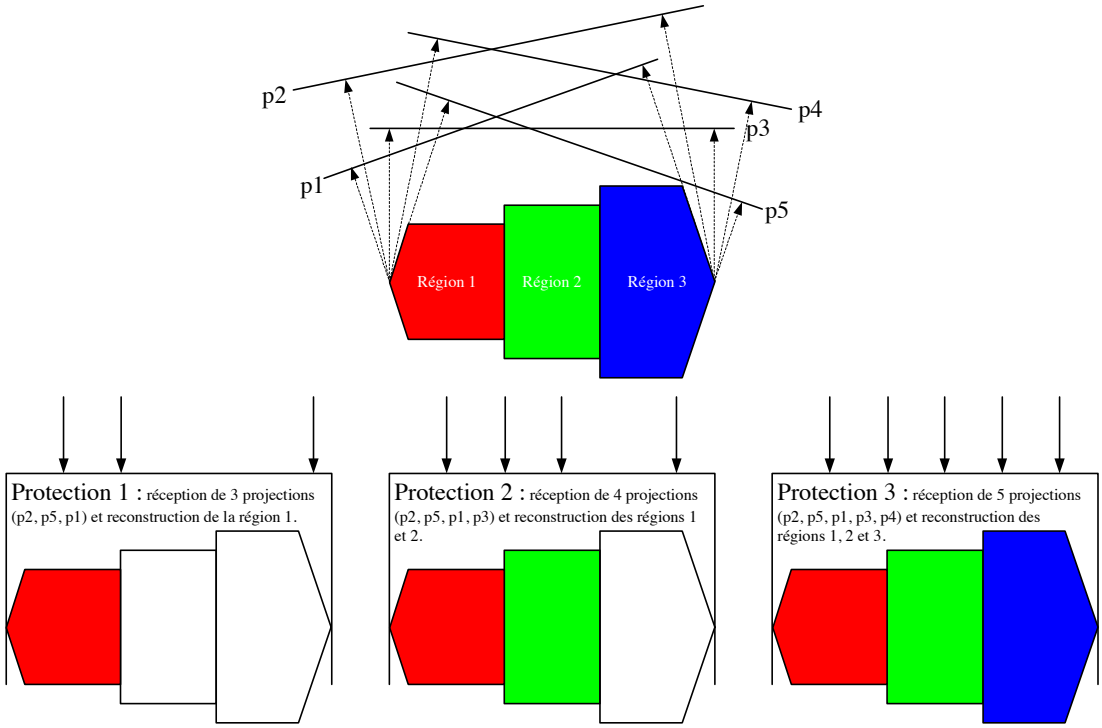


Fig. 5.1: Exemple de la projection d'une mémoire tampon géométrique avec trois niveaux de protections. La réception de trois projections (resp. quatre et cinq) permet la reconstruction de la région 1 (resp. 2 et 3).

sont envoyées dans cinq paquets. Au décodage, on assiste à la reconstruction graduable du support par la transformation Mojette inverse. Avec la réception de moins de trois projections, une quantité négligeable d'éléments d'information (inexploitable dans la représentation de la source) est reconstruit. C'est la réception de trois projections qui permet de reconstruire exactement la région 1 et donc la couche associée. Cette reconstruction est possible quelles que soient les trois projections parmi les cinq envoyées. En outre, l'ordre n'est pas imposé au décodage. Seul le nombre fixe de trois projections définit ce niveau de protection 1. Par ailleurs, la réception de quatre (resp. cinq) projections satisfait les conditions de reconstruction de la région 2 (resp. trois) et de la couche associée. Les niveaux de protection 1, 2 et 3 sont donc définis par trois, quatre et cinq projections nécessaires.

5.2 Choix du support et choix des projections

La reconstruction graduable de la mémoire tampon géométrique peut être réalisée de deux manières distinctes. On peut en effet construire un nombre de mémoires correspondant au nombre de sous-flux à protéger. En appliquant la transformation Mojette sur chacune des mémoires, on obtient pour un angle donné des morceaux de projections que l'on concatène dans un seul paquet. Une deuxième solution consiste à construire une mé-

moire tampon géométrique comprenant l'ensemble des niveaux de protections à apporter. La transformation Mojette est appliquée alors sur la mémoire tampon géométrique entiere. On parle dans ce cas de supports concaténés.
Les considérations du chapitre 4 sur le choix des angles restent valables pour la protection graduable à n niveaux de priorité. En particulier, les angles sont de la forme $(p_i, 1)\ \forall i = 1, ...N$. L'épaisseur du support, pour un nombre de projections m_s nécessaires à la reconstruction du sous-flux s, est donc de m_s ixels (éléments d'information).

5.2.1 Projections concaténées

Construction

La séparation des supports est illustrée à la figure 5.2. Trois supports possèdent des propriétés de reconstruction distinctes pour obtenir trois niveaux de protections graduables. Les sous-flux 1, 2 et 3 sont affectés respectivement dans les supports 1, 2 et 3. Le support 1 (resp. 2 et 3) est reconstructible par deux projections (resp. trois et quatre) prises dans l'ensemble

$$S = \{[-2, 1], [-1, 1], [0, 1], [1, 1]\}.$$

L'épaisseur des supports est en effet en accord avec la contrainte verticale du critère de Katz pour la cas rectangulaire. Simultanément, chaque support est projeté dans les quatres projections. La mise en paquet conduit à la concaténation des morceaux de projections. Le paquet 1 contient alors les projections d'angle $(0, 1)$ de chaque support tout comme le paquet 4 qui contient les projections d'angle $(-2, 1)$ de chaque support.

Expression de la redondance globale

La redondance a été définie (§ 3.3.1 page 61) comme le nombre de bins total sur le nombre d'ixels total moins un soit :

$$Red = \frac{nombre\ total\ de\ bins}{nombre\ total\ d'ixels} - 1.$$

Le nombre de bins pour la projection (p, q) d'un rectangle de dimension $P \times Q$ est donnée par

$$\#bins_{(p,q)} = |q|\,(P-1) + |p|\,(Q-1) + 1. \tag{5.1}$$

Si Q est l'épaisseur du support rectangulaire, et f_s le nombre d'ixels contenus dans le support alors

$$Q = m_s \text{ et } f_s = P \times m_s.$$

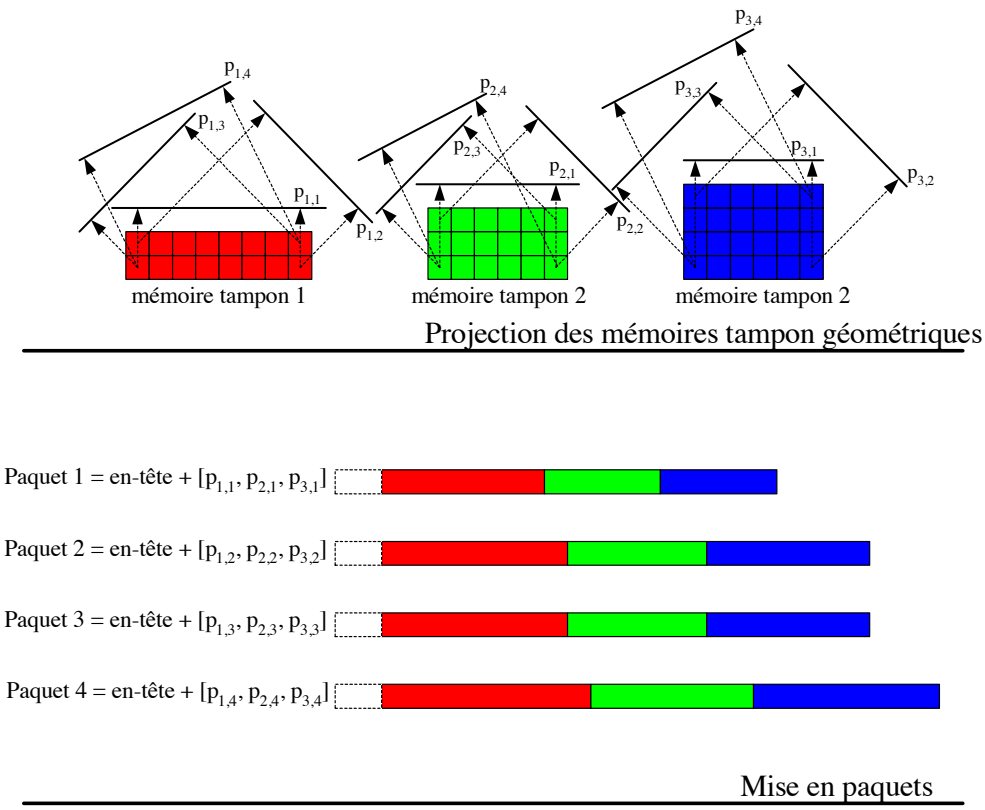


Fig. 5.2: Schéma de protection à trois niveaux de priorité avec séparation des mémoires tampon géométriques. La mise en paquets pour un angle donné s'effectue par concaténation des projections de chaque mémoire. La largeur de la mémoire est fonction de la taille des paquets (contrainte canal).

Le nombre de bins exprimée à l'équation 5.1 devient à $q = 1$, et pour un nombre d'ixels et un niveau de priorité donné :

$$\#bins_{(p,1)} = \frac{f_s}{m_s} + |p|\,(m_s - 1). \tag{5.2}$$

Aussi, le nombre total de bins contenus dans l'ensemble des N projections et des L sous-flux est :

$$\#bins_{N,L} = \sum_{s=1}^{L} \left[\frac{f_s}{m_s} + (m_s - 1) \sum_{i=1}^{N} |p_i| \right]$$

La redondance introduite par la description des L sous-flux s'exprime avec $\sum_{i=1}^{N} |p_i| = \mathcal{P}$ par :

$$\begin{aligned} Red_{N,L} &= \frac{\sum_{s=1}^{L} \left[\frac{f_s}{m_s} + \mathcal{P}(m_s - 1) \right]}{\sum_{s=1}^{L} f_s} - 1 \\ &= \frac{\sum_{s=1}^{L} \frac{f_s}{m_s}}{\sum_{s=1}^{L} f_s} + \mathcal{P} \frac{\sum_{s=1}^{L} (m_s - 1)}{\sum_{s=1}^{L} f_s} - 1. \end{aligned} \tag{5.3}$$

Dans le cas d'un codage MDS, m_s paquets parmi N suffisent à reconstruire un volume source de taille f_s. Aussi, la répartition du volume source dans m_s paquets conduit nécessairement à un nombre de bins par paquet de : $\#bins = \frac{f_s}{m_s}$[27]. Le nombre de bins pour la transformation Mojette donné à l'équation 5.2 exprime un allongement de la taille des descriptions qui se retrouve dans la définition de la redondance globale produite par N projections (deuxième terme de l'équation 5.3). À la différence du code MDS, où tous les symboles redondants sont le résultat de la corrélation de M symboles sources, la transformation Mojette inverse nécessite dans son algorithme un nombre de symboles sources explicites. Cet allongement de taille devient rapidement inacceptable en terme de redondance lorsque le nombre N de descriptions augmente (augmentation de $\mathcal{P}$ dans l'équation 5.3).

Cette première solution permet le transport d'une source graduable par des paquets de même capacité de reconstruction. Le schéma de la figure 5.2 est en outre extensible à autant de niveaux de priorité qu'exige la granularité de la source. Le coût de la redondance associé à chaque coin de mémoire (six dans l'exemple de la figure 5.2) peut être contrebalancé par l'indépendance de projection et la possibilité d'effectuer une reconstruction en parallèle au niveau du décodeur. Les extrémités de chaque projection présentent en effet un nombre d'antécédents univoques qui autorisent le début de la reconstruction Mojette. C'est cette faible corrélation contenue dans les bins extrêmes qui présente cette solution comme sous-optimale. La concaténation des supports avec niveaux de priorité distincts supprime ce sur-débit propre à la séparation.

5.2.2 Supports concaténés

Cette solution est déjà suggérée sur la figure 5.1. L'idée est de concaténer les formes hexagonales présentée au chapitre 4 en une seule mémoire tampon géométrique. L'opérateur Mojette est alors effectué sur le support entier. Les projections obtenues permettent

[27] Sans distinction entre bins sources explicites et bins redondants (produit par la corrélation des symboles source).

une reconstruction graduable du support. Les zones progressivement reconstruites permettent de restituer progressivement les flux multiplexés en fonction de leur degré de pertinence.

Construction

La construction d'une mémoire tampon géométrique s'inspire de la construction présentée au chapitre 4 dans le cas d'une protection non graduable. Des régions qui regroupent des ixels reconstructibles par un même nombre de projections sont construites individuellement pour être ensuite fusionnées en une seule forme à projeter. Sur la figure 5.3, est présentée l'exemple d'un support aux propriétés de reconstruction graduable. De gauche à droite du support, trois régions sont fusionnées et peuvent être reconstruites par trois, quatre et cinq projections prises dans la série

$$S = \{[-2, 1], [-1, 1], [0, 1], [1, 1], [2, 1]\}.$$

De par la forme des deux régions situées aux extrémités du support, la taille des projections est constante. Elle est donnée au chapitre 4 à l'équation 4.11, soit :

$$\#bins = q\Delta k + 1.$$

Ainsi, avec $q = 1,\ \Delta k = 19$, les cinq projections de la figure 5.3 ont une même taille de 20 bins.

Dans le cas des supports concaténés, certains bins sont le résultat de l'addition d'éléments d'information de deux régions à niveaux de priorité distincts. La projection au niveau des frontières provoque une réduction du nombre de bins par rapport à la construction précédente (§ 5.2.1) par les corrélations inter-région introduites.

Nombre d'ixels frontière

La frontière entre deux régions n'est pas aussi franche dans la pratique que le présente la construction précédente. Le niveau m_i de protection affecté à chaque région suppose que toutes les combinaisons de m_i projections reconstruisent une région. Or, en fonction des angles reçus, la frontière entre deux régions n'est pas définie de manière unique. La figure 5.4 montre cette différence pour deux scénarios de réception. La capacité de la région, où deux projections sont nécessaires pour la reconstruction, est plus importante pour la réception 2 que pour la réception 1. Cette différence s'explique par le chevauchement du dilaté des projections reçues sur les deux régions adjacentes .

Pour définir les zones de protection, le scénario de réception le plus défavorable est envisagé. Il s'agit plus précisément de la réception par lequel on minimise le nombre d'ixels reconstruits dans la zone de protection inférieure (m_1 sur la figure 5.4) soit un nombre

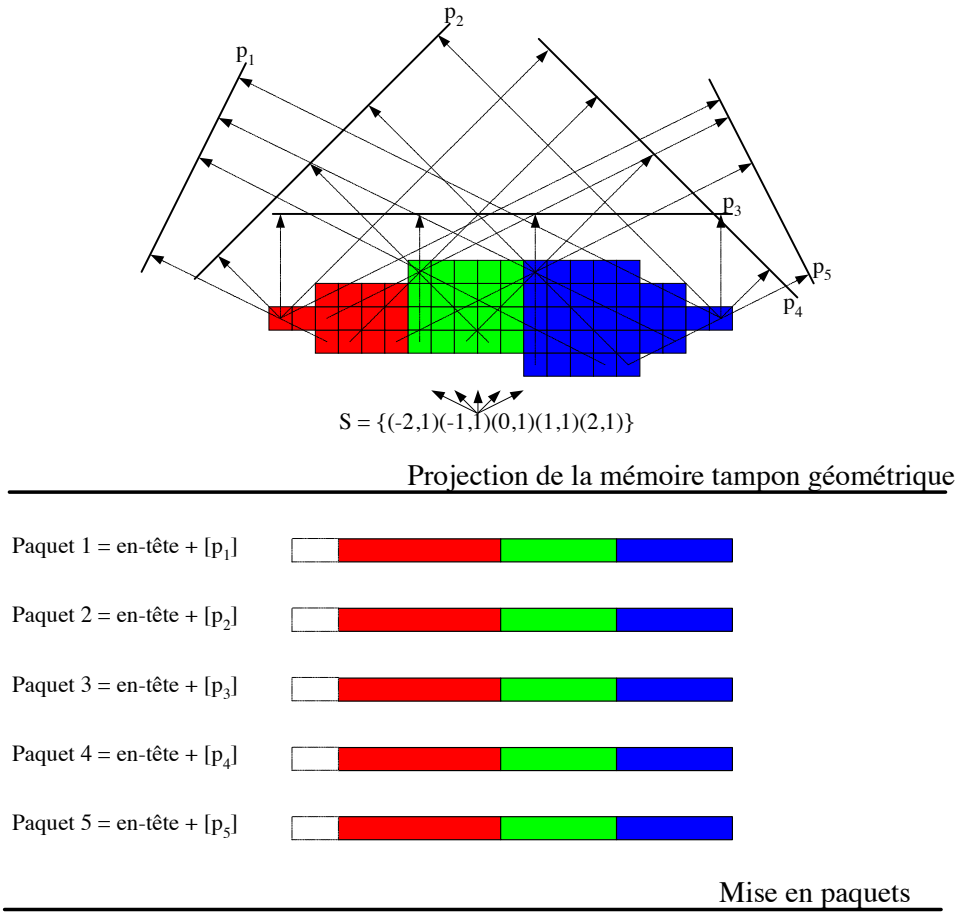


Fig. 5.3: Schéma de protection à trois niveaux de priorité avec concaténation des mémoires tampon géométriques. Il y a dans ce cas correspondance entre un paquet et une projection. Les unités de transport ainsi crées sont de taille constante.

d'ixels frontière le plus important. Ce scénario se traduit par la réception des plus petits angles du secteur $[0,\frac{\pi}{2}]$ (pour $m_1 = 2$, ceux-ci correspondent donc aux deux angles $(p_{\mathsf{max}}, 1)$ et $(p_{\mathsf{max}} - 1, 1)$) et ce parmi l'ensemble de N projections

$$S = \{(p_{\mathsf{max}}, 1), (p_{\mathsf{max}} - 1, 1)...(p_{\mathsf{max}} - N + 1, 1)\}.$$

Ainsi, sur la figure 5.4, la réception 1 effectue la reconstruction de la zone de protection m_1 avec les projections $(3, 1)$ et $(2, 1)$. D'un point de vue algorithmique, l'arrêt de la reconstruction intervient par le manque de correspondance univoque comme le montre la figure 5.5. Sur cette même figure, pour $j = 0$, le nombre d'ixels frontières est de $3 + 2$ soit $p_{\mathsf{max}} + p_{\mathsf{max}} - 1$ avec $p_{\mathsf{max}} = 3$. Pour $j = 1$, ce nombre est de 3 soit égal à p_{max}. Dans le cas général, pour une ligne d'indice j et pour un niveau de protection m_1, le nombre d'ixels

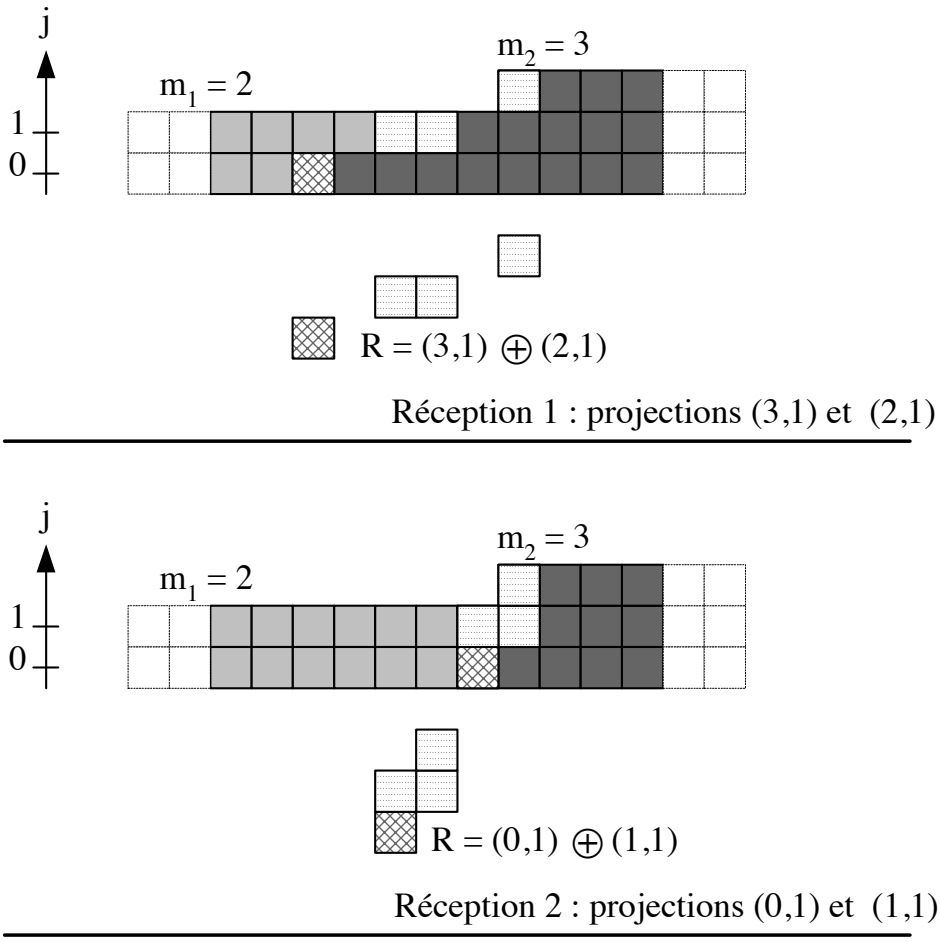


Fig. 5.4: Exemple de frontières entre deux niveaux de protection pour deux scénarios de réception. La dilatation des éléments structurants pour les deux scénarios délimite les deux régions.

appartenant à la zone de protection supérieure m_2 avec $m_2 = m_1 + 1$ est

$$\begin{aligned} \#ixels_j &= \sum_{i=0}^{m_1-j-1} p_{\max} - i \\ &= (m_1 - j)p_{\max} - \sum_{i=0}^{m_1-j-1} i \\ &= (m_1 - j)p_{\max} - \frac{(m_1 - j - 1)(m_1 - j)}{2}. \end{aligned} \tag{5.4}$$

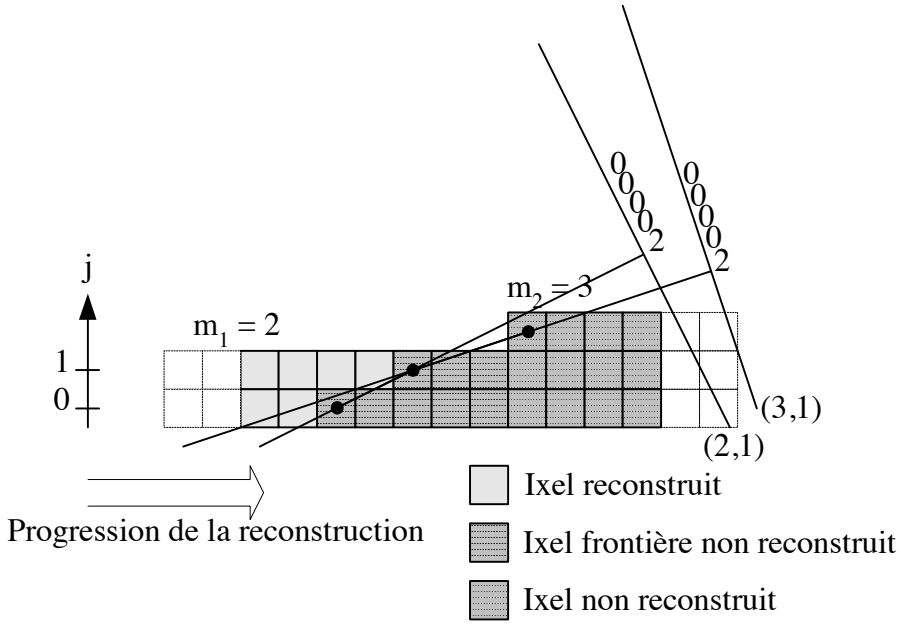


Fig. 5.5: Arrêt de la reconstruction de la zone protégée m_1 par manque de correspondance univoque.

Le nombre total d'ixels frontières sur les m_1 lignes appartenant à la zone m_2 est donc de

$$\begin{aligned}
\#ixels_{frontière1-2} &= \sum_{j=0}^{m_1-1} [(m_1-j)p_{\max} - \frac{(m_1-j-1)(m_1-j)}{2}] \\
&= \sum_{j=0}^{m_1-1} [m_1 p_{\max} - \frac{m_1(m_1-1)}{2} + j(-p_{\max} + \frac{m_1+m_1-1}{2}) - \frac{j^2}{2}] \\
&= m_1^2 p_{\max} - \frac{m_1^2(m_1-1)}{2} + \frac{(m_1-1)m_1}{2}[-p_{\max} + m_1 - \frac{1}{2}] - \frac{(m_1-1)m_1(2m_1-1)}{12} \\
&= -\frac{m_1^3}{6} + \frac{m_1^2 p_{\max}}{2} + m_1(\frac{p_{\max}}{2} + \frac{1}{6}) \\
&= -\frac{m_1}{6}(m_1+1)(m_1-3p_{\max}-1).
\end{aligned}$$

Exemples :

- Dans l'exemple de la figure 5.4, où $p_{\max} = 3$ et $m_1 = 2$
  $\#ixels_{frontière1-2} = 8$.
- Si $p_{\max} = 3$ et $m_1 = 3$ alors
  $\#ixels_{frontière1-2} = 14$.
- Si $p_{\max} = 6$ et $m_1 = 8,\ m_2 = 9,\ m_3 = 10$ alors
  $\#ixels_{frontière1-2} = 132$ et $\#ixels_{frontière2-3} = 450$.

Le nombre d'ixels frontière n'est pas fonction du nombre d'ixels contenus dans chaque région. L'effet "frontière" (nombre d'ixels frontière important) sera cependant atténué

dans le cas de régions de grande capacité. Le compromis dimension de la mémoire tampon géométrique et taille des projections (contraintes de temps réel) est ici toujours valable.

5.3 Autre construction de mémoire tampon géométrique pour les supports concaténés

Les restrictions du paragraphe 5.2.2 imposent une protection avec des niveaux consécutifs qui se concrétisent dans la construction de la mémoire tampon géométrique par une marche de un ixel entre deux régions distinctes soit $m_1 = m_2 - 1$. Néanmoins l'application, et de manière plus générale la source, peuvent nécessiter une protection très différente entre ses sous-flux. Par exemple, la décomposition en sous-bandes d'une image fait appel a priori à des niveaux de protection très inégaux [Moreau 95]. On présente dans ce paragraphe une nouvelle construction de supports permettant d'affecter des niveaux de protection non consécutifs aux sous-flux applicatifs. Cette amélioration fournit une plus grande souplesse pour respecter la fonction de priorité de la source. L'optimisation de la capacité du support est tout d'abord présentée. La construction de zones de niveaux protection non consécutifs permet en effet d'ajouter des éléments source dans la mémoire tampon. Cette modification de la forme de la mémoire implique un nouveau calcul du nombre d'ixels frontière qui est exposé dans un deuxième temps. La construction finale de mémoires tampon symétriques est détaillée dans un troisième temps.

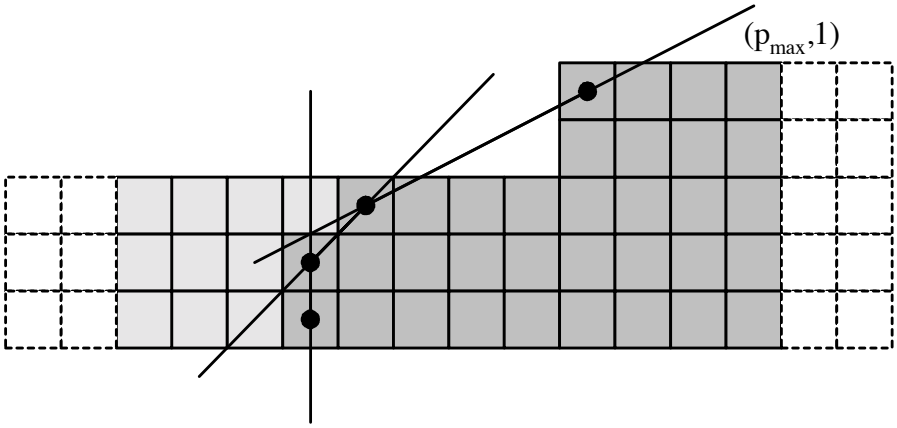
5.3.1 Optimisation de la capacité du support

Le support 1 de la figure 5.6 présente une reconstruction partielle par trois projections (région de gauche) et une reconstruction totale avec cinq projections soit $m_2 = m_1 + 2$. Le sous-ensemble suffisant considéré est composé des projections (2,1), (1,1) et (0,1) correspondant au scénario le plus défavorable lors de l'envoi de l'ensemble

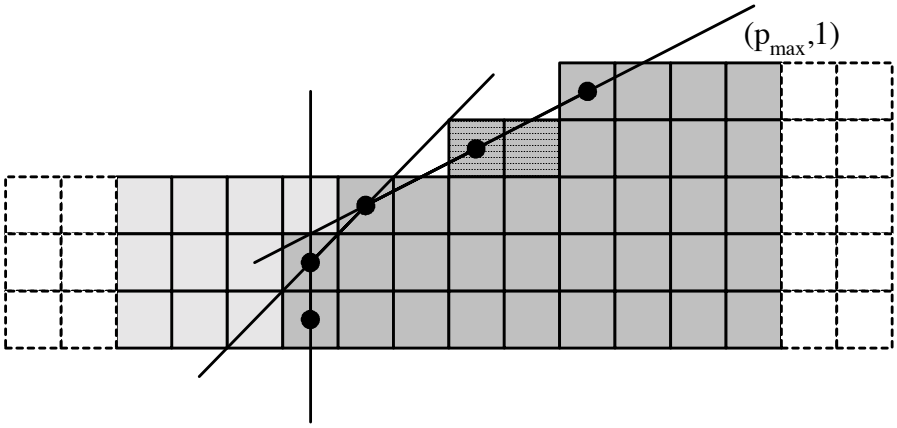
$$S = \{(-2,1),(-1,1),(0,1),(1,1),(2,1)\}.$$

La reconstruction partielle s'achève par le manque de correspondance univoque dans le sous-ensemble de projections. Les bins bloquants la reconstruction sont le résultat de la somme de deux ixels. On dénombre en outre trois équations à quatre inconnues (représentées par des points noirs sur la figure 5.6). L'ajout des deux ixels sur le support 2 (en hachure sur la figure 5.6) ne compromet pas la reconstruction totale puisque les bins décrivant la région de droite (reconstructible par cinq projections) sont de manière générale le résultat de la somme de cinq ixels dans les directions de projection. Ces deux ixels supplémentaires modifient uniquement le nombre d'ixels antécédents dans le bin bloquant de la direction (2,1) soit $(p_{\mathsf{max}}, 1)$, passant de deux à trois. Autrement dit, l'optimisation de la capacité du support porte sur la maximisation du nombre total d'ixels. Elle revient

simplement à concevoir des mémoires tampon pq-convexes pour toutes les droites discrètes définies par les directions de projection du scénario de réception le plus défavorable. C'est donc ce scénario qui contraint l'optimisation de part les ixels qu'il reconstruit.



Support 1 et réception des projections (2,1), (1,1) et (0,1)



Support 2 et réception des projections (2,1), (1,1) et (0,1)

Fig. 5.6: Optimisation de la capacité du support. L'ajout des deux ixels sur le support 2 par rapport au support 1 ne modifie pas les possibilités de reconstruction (reconstruction totale par cinq projections).

Dès lors, en retenant la construction du support 2 de la figure 5.6, les ixels de coordonnées (k,l) seront considérées comme appartenant au support B si ils sont sur la droite

ou en deçà de la droite $m = -k + p_{\mathsf{max}} l$. On a donc

$$X_i \left| \begin{array}{l} k \\ l \end{array} \right. \in B \Leftrightarrow m \ \mathsf{6} \ -k + p_{\mathsf{max}} l.$$

5.3.2 Nombre d'ixels frontière

De cette manière, on ramène le dénombrement des ixels frontière avec $m_2 > m_1 + 1$ au cas d'un décrochement d'un ixel entre deux régions de niveau de protection différent soit $m_2 = m_1 + 1$ (§ 5.2.2). Compte tenu de la valeur de q qui est égale à 1, le décrochement entre les deux régions d'épaisseur $m_1 q$ et $m_2 q$ ne peut être que de un ixel.
La figure 5.7 décompose ainsi les ixels frontière sous deux types : les ixels frontières dénombrés au paragraphe 5.2.2 (de type 1) et les ixels frontières dénombrés dans une forme géométrique triviale (de type 2). Les ixels de type 2 sont en effet contenus dans un trapèze. Soit I_2, le nombre de ce type d'ixel contenus dans un trapèze, on a

$$I_2 = \frac{1}{2}(m_2 - m_1)(m_2 - m_1 - 1)p_{\mathsf{max}} + m_1(m_2 - m_1 - 1)p_{\mathsf{max}}. \tag{5.5}$$

L'exemple de la figure 5.7 fait état d'une protection à deux niveaux. La région 1 est reconstructible par trois projections soit $m_1 = 3$ et la région 2 est reconstructible par sept projections soit $m_2 = 7$. L'ensemble des projections considéré est

$$S = \{(-3,1), (-2,1), (-1,1), (0,1), (1,1), (2,1), (3,1)\},$$

soit $p_{\mathsf{max}} = 3$. L'application numérique de 5.5 révèle ainsi 45 $(= 27 + 18)$ ixels de type 2.

D'une manière similaire au paragraphe 5.2.2, on peut dénombrer les ixels de type 1. Soit I_1, le nombre d'ixels de type 1, on a pour l'exemple de la figure 5.7 avec $j \in \{0, 1, 2\}$

$$\begin{aligned} I_1 &= 3 + 5 + 6 \\ &= 14. \end{aligned}$$

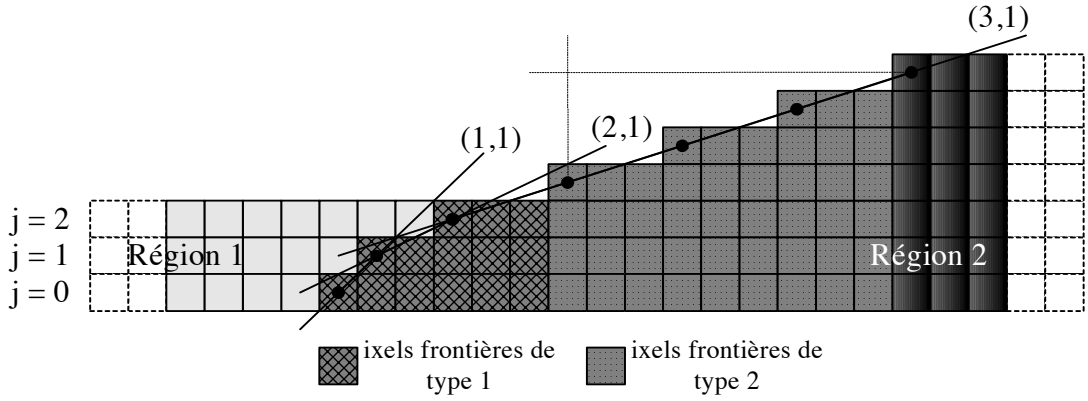


Fig. 5.7: Détails du comportement à la reconstruction des ixels situés à la frontière de deux régions. La région 1 est reconstructible par trois projections et la région 2 est reconstructible par sept projections.

Le nombre d'ixels frontière total se décompose par la somme des deux types soit

$$\#ixels_{frontière1\rightarrow2} = I_1 + I_2.$$

Dans l'exemple présenté, on dénombre ainsi 59 ixels frontières.
Ce nombre d'ixels s'ajoute aux ixels de la région 2 ayant pour niveau de protection m_2 projections nécessaires pour sa reconstruction totale.

5.3.3 Construction finale

Dans l'objectif de minimiser la redondance, il convient de maximiser les corrélations contenues dans chaque bin. Pour être plus précis, il convient de **maximiser le nombre d'antécédents** dans chaque bin. Ce nombre maximal correspond au niveau de protection de la région considérée. Il est nécessaire néanmoins pour entreprendre l'algorithme Mojette inverse que des correspondances univoques soient contenues dans les projections du sous-ensemble suffisant. On rappelle ici que c'est cette contrainte qui confère à la transformation Mojette la propriété $(1+\varepsilon)$MDS.
Le décrochement de la région 2 par rapport à la région 1, figure 5.7, est coûteux par le nombre de bins qui le décrivent. Il suffit pour réduire la transition de concaténer les deux régions de manière symétrique comme représenté figure 5.8. Mais dans ce cas, une étude particulière de la transition est nécessaire car le scénario qui maximise le nombre d'ixels frontière n'est plus limité à la réception des angles avec les plus grands p_i pour $i = 1, 2, ...M$ majorés par p_{max}. La réception des plus petits p_i pour $i = 1, 2, ...M$ minorés par p_{min} influe en effet sur la reconstruction des ixels frontières.

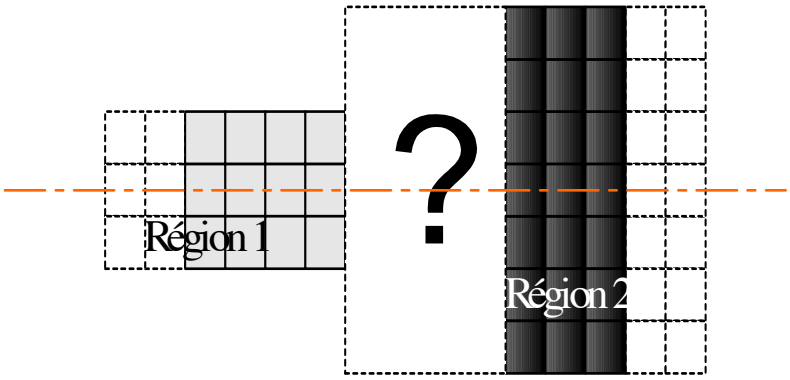


Fig. 5.8: Concaténation des supports par rapport à un axe. La région de transition est à repréciser.

Les transitions sont suffisamment douces *i.e* l'angle $(p_{\mathsf{max}}, 1)$ est proche de 0 pour que le décrochement ne soit que de un ixel (voir figure 5.9). Quel que soit le niveau de protection m_2, on ne considère que les décrochements de un ixel de part et d'autre de l'axe de symétrie. Un calcul complémentaire analogue à celui présenté dans le paragraphe précédent permet de dénombrer les ixels frontières contenus dans une forme géométrique triviale. En revanche, on doit préciser à nouveau le nombre d'ixels frontière de type 1.

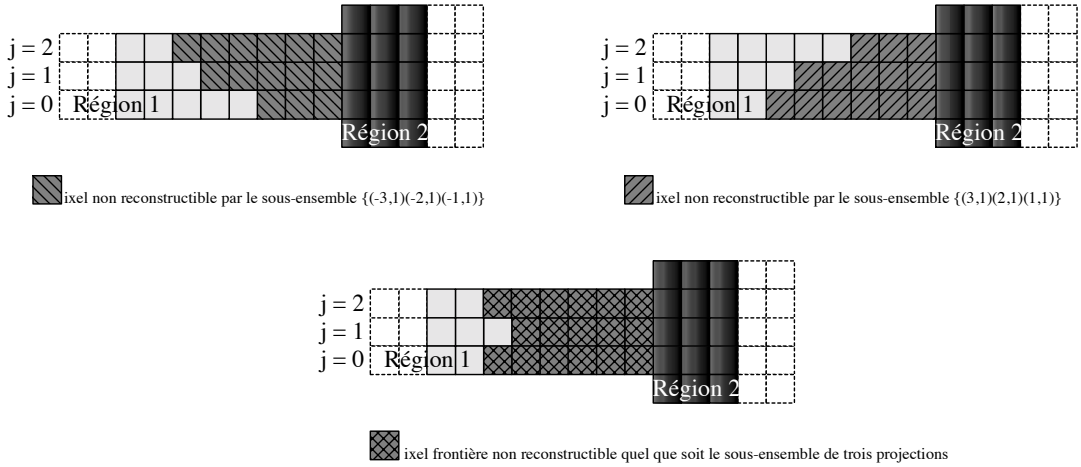


Fig. 5.9: Résultat de la réunion des deux ensembles d'ixels frontière non reconstructibles par les deux sous-ensembles de trois projections {(-3,1) ; (-2,1) ; (-1,1)} et {(3,1) ; (2,1) ; (1,1)}.

Ce nombre d'ixels frontière est le résultat de la réunion des deux ensembles d'ixels frontière non reconstructibles par les deux sous-ensembles de projections défavorables définis précédemment ({(3,1), (2,1), (1,1)} et {(-3,1), (-2,1), (-1,1)} sur la figure 5.9). Notons E_1 et E_2, ces deux ensembles d'ixels non reconstructibles. Pour E_1, le nombre d'ixels pour

une ligne j est donné à l'équation 5.4 à savoir

$$\#ixels_{j,E_1} = (m_1 - j)p_{\mathsf{max}} - \frac{(m_1 - j - 1)(m_1 - j)}{2}. \tag{5.6}$$

De manière similaire, on dénombre pour E_2 et pour une ligne j

$$\begin{aligned} \#ixels_{j,E_2} &= \sum_{i=0}^{j}(\text{-}p_{\mathsf{min}} - i) \\ &= \text{-}p_{\mathsf{min}}(j+1) - \frac{j(j+1)}{2}. \end{aligned} \tag{5.7}$$

Dénombrer le nombre d'ixels frontière revient alors à définir à partir de quelle ligne j l'ensemble est le plus conséquent en nombre d'ixels. Soit j, la ligne à partir de laquelle l'ensemble E_1 devient plus conséquent que l'ensemble E_2, on a

$$\begin{aligned} (m_1 - j)p_{\mathsf{max}} - \frac{(m_1 - j - 1)(m_1 - j)}{2} &> \text{-}p_{\mathsf{min}}(j+1) - \frac{j(j+1)}{2} \\ j(m_1 - p_{\mathsf{max}} + p_{\mathsf{min}}) &> \frac{m_1^2}{2} - m_1(p_{\mathsf{max}} + \frac{1}{2}) - p_{\mathsf{min}}. \end{aligned} \tag{5.8}$$

Le problème revient à déterminer le point d'intersection entre deux paraboles en j^2. On distingue alors deux cas : le cas symétrique (exemple figure 5.9) du cas non symétrique, où N, le nombre de projections total, est respectivement impair et pair.

Cas 1 : N impair $\Leftrightarrow p_{\mathsf{min}} = \text{-}p_{\mathsf{max}} \Leftrightarrow N = 2p_{\mathsf{max}} + 1$

Éq.5.8 devient

$$\begin{aligned} j(m_1 - 2p_{\mathsf{max}}) &> p_{\mathsf{max}}(1 - m_1) - \frac{m_1}{2}(1 - m_1) \\ j(m_1 - 2p_{\mathsf{max}}) &> \text{-}\frac{m_1 - 2p_{\mathsf{max}}}{2}(1 - m_1). \end{aligned}$$

Pour un décrochement de 1 ixel, on a $m_1 \leqslant m_2 - 2$ et dans le cas où N projections reconstruisent totalement les niveaux m_1 et m_2, on a $m_2 \leqslant 2p_{\mathsf{max}} + 1$ d'où

$$m_1 \leqslant 2p_{\mathsf{max}} - 1 \Leftrightarrow j \leqslant \frac{m_1 - 1}{2}.$$

Cas 2 : N pair $\Leftrightarrow p_{\mathsf{min}} = \text{-}p_{\mathsf{max}} + 1 \Leftrightarrow N = 2p_{\mathsf{max}}$

Éq.5.8 devient

$$j(m_1 - 2p_{\mathsf{max}} + 1) > \frac{m_1^2}{2} - \frac{m_1}{2}(2p_{\mathsf{max}} + 1) + p_{\mathsf{max}} - 1$$

$N = 2p_{\mathsf{max}} \Leftrightarrow m_1 \leqslant 2p_{\mathsf{max}} - 2$, d'où

$$j \leqslant \frac{\frac{m_1^2}{2} - \frac{m_1}{2}(2p_{\mathsf{max}} + 1) + p_{\mathsf{max}} - 1}{m_1 - 2p_{\mathsf{max}} + 1}.$$

Le calcul des ixels appelés frontière est indispensable dans le multiplexage sur le support des éléments d'information source à protéger. Chaque position de la mémoire tampon est reconstructible de manière déterministe par un nombre de projections en correspondance avec le niveau de protection et ce quel que soit le scénario de réception. La figure 5.10 illustre le niveau de reconstruction à chaque arrivée de projection et ce pour tout les scénarios de réception possible (ordre d'arrivée des projections). Le support correspondant est reconstructible graduellement par onze, quatorze, quinze puis seize projections parmi seize envoyées. Les écarts types entre scénarios ne présentent pas de différences notables dans le nombre d'ixels reconstruits.

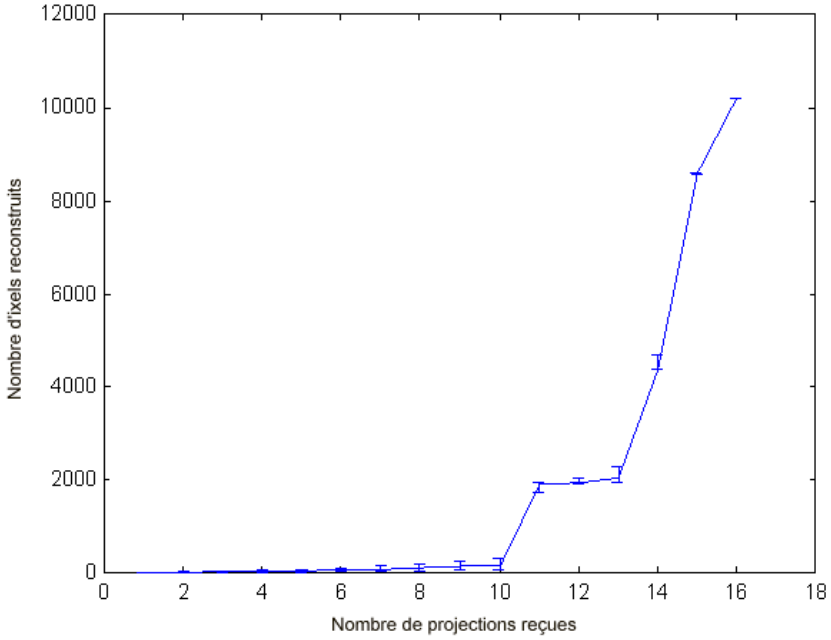


Fig. 5.10: Nombres d'ixels reconstruits en fonction du nombre de projection reçues et de leur ordre d'arrivée.

On se propose d'effectuer dans la section suivante une première application de ce codage à description multiple pour une source JPEG progressif.

5.4 Application à la transmission d'image

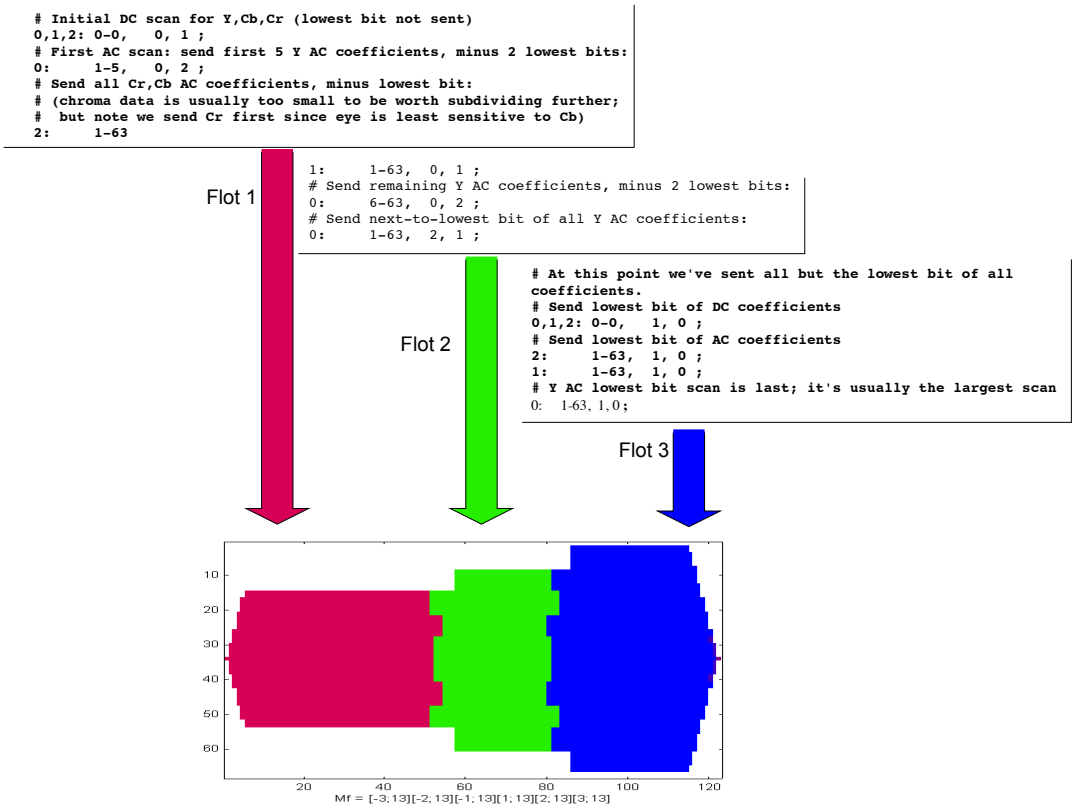


Fig. 5.11: Multiplexage de trois sous-flux JPEG sur une mémoire tampon géométrique à trois niveaux de protection.

Une première application de la reconstruction graduable d'une mémoire tampon géométrique présentée ci-dessus est extraite de [Parrein 01a]. La source est un codeur JPEG progressif [Group 98] décomposée en trois sous-flux dont un s'apparente à la version de base de la source et les deux autres sont utilisés en couches de raffinement. Une description complète des flux et de la mémoire tampon géométrique utilisée pour le multiplexage est donnée à la figure 5.11. Le codage prioritaire est paramètré pour que le sous-flux 1 ait une reconstruction exacte à partir de la réception de trois projections parmi six soit un effacement de paquets de 50 % autorisé. Les sous-flux 3 et 4 sont respectivement reconstruits par quatre et cinq projections parmi six envoyées. Leur taille est approximativement constante avec environ 1580 bins pour chaque projection. Le volume d'information initial est composé de 5645 octets et les six projections représentent 9480 octets, soit une redondance de 35 % sans aucune compression des projections. La redondance mesurée est du même ordre de grandeur que celles de [Goyal 98b] et [Jiang 99].
Les images obtenues pour un canal à effacement sont données à la figure 5.12. La reconstruction de gauche est réalisée après la perte de 50 % des descriptions, correspondant à un

débit source de 0,45 bpp (bit par pixel pour l'image finale) alors que le débit à la réception est de 1,12 bpp. L'image du centre utilise quatre projections parmi six pour un débit source efficace de 0,79 bpp, comparé au 1,49 bpp de débit à la réception. Enfin, l'image de droite correspond à la reconstruction complète à 1,33 bpp comparé au débit de 1,89 bpp (provoqué en grande partie par la projection supplémentaire).

Les volumes initiaux de chaque sous-flux sont de 1910, 1413 et 2322 octets pour les sous-flux 1, 2 et 3. La mise en œuvre MDS de la même protection fournit 637 ($= \frac{1910}{3}$), 354 ($= \frac{1413}{4}$) et 465 ($= \frac{2322}{5}$) éléments dans chaque mot code pour une longueur totale de 1456 éléments. En comparaison, l'approche Mojette distribue l'information dans des paquets avec 8 % d'éléments en sus. Cependant, cet allongement de la taille des paquets peut être réduit à 1 % par l'élongation du support avec des éléments d'information binaires (§ 4.2.2).

Fig. 5.12: Résultats de la protection Mojette pour l'image Rose codée en JPEG progressif à 1,33 bpp. De gauche à droite, le nombre de paquets perdu est de 3, 2 et 1 parmi 6 pour un débit de 0,45 bpp, 0,79 bpp et 1,33 bpp.

5.5 Conclusion

Nous avons proposé dans ce chapitre d'effectuer un codage prioritaire en utilisant les propriétés de reconstruction d'un support par des projections discrètes. La progression de la reconstruction est assurée par l'arrivée successive des projections, indépendamment de l'ordre de réception. On reconstruit par ce mécanisme l'information de haute priorité en premier pour ensuite reconstruire les couches de raffinements qui constituent la source originale. Nous avons ainsi présenté la construction d'une mémoire tampon géométrique à reconstruction graduable, détaillé les choix des supports et des angles Mojette, et appliqué ce codage à la transmission par paquets d'une source JPEG progressif. Au terme de cette étude, la concaténation des mémoires tampon hexagonaux présentées au chapitre 4 est apparue comme la solution optimale. On insiste sur le fait que l'optimisation est effectuée sur le scénario de réception où le nombre d'ixels frontière est maximisé. C'est pour ce scénario que sont dessinées les régions de protection graduable. Un nombre de projections donnée reconstruira exactement le même nombre d'ixels et ceux-ci ont une position sur le support invariante quel que soit le scénario de réception. Il est dans cas nécessaire de dénombrer l'appartenance des ixels appelés frontières entre deux régions limitrophes. La figure **??**

résume les résultats de ce chapitre pour une protection à trois étages : reconstruction progressive par trois, cinq et sept projections parmi l'ensemble de sept projections

$$S = \{(\text{-}3,1),(\text{-}2,1),(\text{-}1,1),(0,1),(1,1),(2,1),(3,1)\}.$$

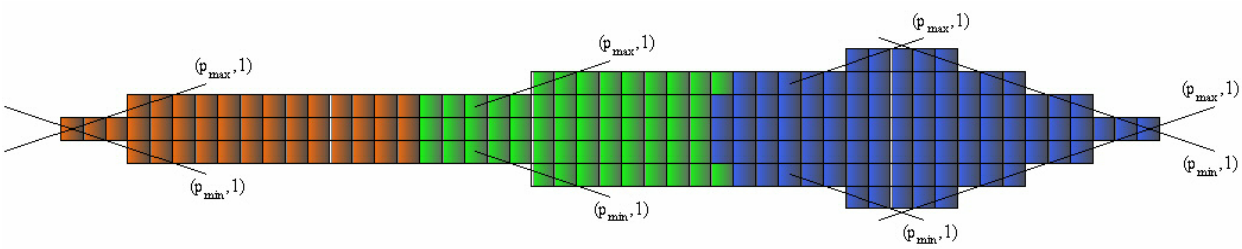


Fig. 5.13: Présentation d'une mémoire tampon géométrique contenant trois niveaux de protections. La région de gauche (resp. centrale et de droite) est reconstructible par trois (resp. cinq et sept) projections parmi sept.

La nature progressive de la reconstruction et la répartition de l'information source dans des projections discrètes confèrent au codage présenté des propriétés de description multiple. La taille constante des descriptions générées décrit en outre un mécanisme équilibré. Dès lors, si des pertes surviennent pendant le transport par paquets de la source, celle ci verra le volume des informations de basse priorité se dégrader en premier. Les projections ayant même valeur dans la reconstruction, ce codage ajoute de la robustesse à la source lors de sa transmission sur réseaux sans contrôle d'intégrité extensible.
Le niveau de protection affecté à chaque flux est cependant arbitraire (§ 5.4). Il convient pour optimiser la redondance globale allouée dans la protection de considérer les propriétés de la source et du canal de manière à proposer un codage conjoint source-canal.

Chapitre 6

Description multiple de l'information par transformation Mojette

Sommaire

6.1 Propriétés de description multiple 120

6.2 Optimisation du système de protection inégale 121

6.2.1 Allocation optimale de la redondance 121

6.2.2 À propos de la qualité 124

6.2.3 Voies de retour canal 133

6.3 Protection égale et protection inégale 135

6.3.1 Débit constant 135

6.3.2 Débit variable 138

6.3.3 Effet de pertes 139

6.4 Conclusion 141

Les deux chapitres précédents se sont axés sur la construction de mémoires tampon géométriques projetables dans les cas de protection égale et inégale (EEP, UEP[28]). Les résultats ont de plus démontré les caractéristiques presque MDS de ce codage à description multiple. Ce chapitre prétend confronter ces deux formes d'utilisation de la redondance. Dans le premier cas, la redondance est introduite de manière uniforme pour l'ensemble des données. L'arrivée d'un nombre fixé de m descriptions autorise la reconstruction de la source. En dessous de ce nombre, aucun élément source n'est délivré aux couches supérieures (ou très peu dans le cadre de la transformation Mojette). Au delà du seuil de reconstruction, les descriptions n'apportent pas plus d'information. Par les deux niveaux

[28] EEP : Equal Error Protection
UEP : Unequal Error Protection

de qualité qu'elle implique (tout ou rien), la protection égale ne possède pas la progressivité que permettent les codes à description multiple.
Dans le deuxième cas, la redondance est fonction de l'importance du sous-flux qu'elle protège. Un vecteur m de dimension correspondant aux nombres de sous-flux qui décrivent la source révèle alors le nombre de descriptions nécessaires aux décodages de chacun. [Albanese 96] évoque pour ce vecteur le terme de fonction de priorité. On parle dans ce cas de description multiple par le raffinement que se fournissent les descriptions entre elles.
Pour les deux cas, il convient d'allouer cette redondance de manière combinée en fonction des propriétés de la source qui dicte l'importance du sous-flux et en fonction des propriétés du canal qui indique le taux d'erreurs possibles aux cours de la transmission.
Dans une première section, nous mettrons en évidence les propriétés de description multiple de la protection inégale proposée au chapitre 5. Une deuxième section apporte une solution d'optimisation pour ce système de protection (*PET system*). La dernière section est consacrée à la comparaison des deux codages à protection égale (EEP) et à protection inégale (UEP).

6.1 Propriétés de description multiple

La protection proposée au chapitre 5 est extensible à autant de niveaux que l'exige la granularité de la représentation source. Quel que soit le choix du support (§ 5.2), l'ajout d'un niveau de protection ne conduit pas à une augmentation de la complexité (fonction linéaire du nombre d'éléments d'information et du nombre de projections). Le codage est une généralisation du problème de description multiple à N canaux où chaque projection est considérée comme une description équivalente et partielle de la source.
Chaque couche s de la source est caractérisée par m_s, le nombre de projections qui la reconstruisent exactement. Les distorsions latérales sont donc égales (§ 2.2). La qualité de la reconstruction est dépendante uniquement du nombre de descriptions et non des descriptions spécifiques qui sont utilisées. En outre, selon le choix du support la taille des projections peut être constante. Cette propriété engendre donc un débit constant pour chaque description. L'égalité des distorsions latérales et des débits engendrés par chaque description confère au codage les deux propriétés de description multiple équilibrée (BMDC[29] voir [Jiang 99]). Or, notre objectif est de transmettre une source graduable sur un réseau IP où aucune priorité n'est traitée. Tout les paquets de même taille sur ce type de réseau ont a priori une probabilité égale d'être effacés. Par conséquent, notre codeur, produisant des paquets de taille identique et de même importance, est mieux adapté pour cette application que les codeurs classiques basés sur une analyse multirésolution.
D'autre part, le nombre de descriptions est élevé dans cette application car la taille des paquets est limitée. Par exemple, sur Ethernet, la taille du paquet IP le plus large trans-

[29] Balanced Multiple Description Coding

missible (MTU)[30] est de 1500 octets. Supposons que nous voulons transmettre une image par deux descriptions de 3000 octets chacune. Bien que les applications aient la possibilité de transmettre des paquets plus larges que la limite MTU, le protocole IP fragmente les deux gros paquets en plusieurs petits, les transmet séparément et les réassemble à la réception. Cependant si un seul de ces petits fragments est supprimé, le paquet entier donc la description partielle est inutilisable. C'est aujourd'hui un des problèmes des serveurs vidéo du marché qui segmentent les données émises en gros blocs (*e.g.* 16 koctets). Pour prévenir cet effet "domino", il est préférable de décrire la source par quatre descriptions de 1500 octets chacune.

6.2 Optimisation du système de protection inégale

L'algorithme PET original [Albanese 96] n'inclut pas la résolution des niveaux de protection. Ces niveaux sont définis de manière plus ou moins arbitraire par l'utilisateur comme nous l'avons fait pour la première application (transmission d'un jpeg progressif) présentée au 5.4. Leicher [Leicher 94], en appliquant le système PET à des flux MPEG, spécifie de cette manière une fonction de priorité avec un taux de 60% de paquets nécessaires à la reconstruction des trames I. Les trames B et P demande ensuite respectivement 80% et 95% des paquets pour leurs restitutions. Ces taux apportent une protection superflue si le canal présente peu d'effacements. En outre, si une visualisation grossière est requise le taux de trame B peut diminuer au détriment du taux des trames P. Le choix de la protection est ainsi en forte dépendance avec la composition de la source et l'état du canal. Les deux paramètres que sont la qualité et la perte de paquets sont fusionnés par un critère qui utilise l'espérance mathématique de la qualité, critère d'optimisation retenu dans l'allocation optimale de la redondance.

6.2.1 Allocation optimale de la redondance

En regroupant les deux paramètres constituant la source et le canal dans l'espérance mathématique de la qualité, on se propose ici de calculer le nombre de projections (de paquets) nécessaires à la reconstruction d'une série de flux source. Un faible nombre de projections constituera pour un flux une protection élevée. À l 'inverse, une faible protection se traduira par un nombre de projections proche de N, le nombre de projections envoyées.

Formalisme d'optimisation

On suppose un codeur en couche en amont du système Mojette de protection qui délivre une série de L flux graduables. Pour tout $s = 1, 2...L$ et pour tout $k = 1, 2...K_s$,

[30] Maximum Transmission Unit

$f_{s,k}$ représente le $k^{ième}$ élément du flux s. La structure en couche implique que l'ensemble f_s n'est utilisable que si f_{s-1} est disponible.

Soit ρ_s, la fonction de priorité à calculer qui indique pour un flux s, le nombre de projections nécessaires à sa reconstruction. La structure en couche implique que ρ est une fonction monotone croissante i.e. :

$$\rho_1 \leq \rho_2 \leq ... \leq \rho_L.$$

Soit Q_s, la qualité estimée au décodage du flux s. Par la graduabilité de la source, on a nécessairement $Q_s \geq Q_{s-1}$. On peut définir un incrément de qualité q_s positif ou nul par

$$q_s = Q_s - Q_{s-1}.$$

Pour déterminer la fonction de priorité ρ, on utilise une estimation du profil des pertes du canal. Cette estimation est donnée par probabilité p_i ; $i = 0, 1...N$, telle que p_i est la probabilité de perdre exactement i paquets. On estime que la probabilité de recevoir au moins le flux s est donnée par la fonction de distribution cumulée $c(\rho_s) = \sum_{i=0}^{N-\rho_s} p_i; \rho_s = 1, 2...N$ comme étant la probabilité de perdre au plus $k = N - \rho_s$ paquets.

Soit X la variable aléatoire représentant le nombre de paquets reçus, ρ_s le nombre de paquets nécessaires pour recevoir le flux s. L'espérance mathématique de la qualité est donnée par :

$$E(Q) = \sum_{s=0}^{L} Q_s P[\rho_s \leq X < \rho_{s+1}]. \tag{6.1}$$

Si $Q_s = \sum_{j=0}^{s} q_j$, on a

$$\begin{aligned} E(Q) &= \sum_{s=0}^{L} \left(\sum_{j=0}^{s} q_j \right) P[\rho_s \leq X < \rho_{s+1}], \\ &= \sum_{j=0}^{L} \sum_{s=j}^{L} q_j P[\rho_s \leq X < \rho_{s+1}], \\ &= \sum_{j=0}^{L} q_j P[X > \rho_j], \\ &= \sum_{j=0}^{L} q_j c(\rho_j). \end{aligned} \tag{6.2}$$

Afin de déterminer la fonction de priorité ρ, on maximise $E(Q)$ fonction de la densité de probabilité p_i. L'incrément de qualité peut être calculé simplement à partir de la différence (entre la reconstruction des flux s et $s-1$) des PSNR, des erreurs quadratiques moyennes ou encore à l'aide de critères perceptuels de qualité (note objective à partir d'évaluations subjectives de la qualité).

Algorithme pour le calcul de la fonction de priorité

L'algorithme présenté détermine la fonction de priorité, à savoir pour chaque flux le nombre de paquets nécessaires à leur reconstruction. La description multiple s'effectue sur une source de taille réduite (sortie d'un schéma de compression). Aussi une recherche exhaustive des nuplets qui maximisent l'espérance mathématique de la qualité est envisagée. Comme il est supposé dans la partie précédente, on contraint $\rho(i) \leq \rho(i+1)$. De plus, on suppose que la perte d'un élément d'information issu d'un flux binaire progressif entraine la perte des éléments à venir. La recherche se termine dès lors qu'un seuil de qualité est atteint **ou** que la capacité du canal est dépassée. Il est à noter enfin que le calcul du nombre de projections nécessaires se fait en supposant que les paquets ont une charge utile "MDS". Le surplus d'informations nécessaires à l'inversion de la transformation Mojette (paramètre ε) est négligeable. À savoir si lp mesure la longueur d'un paquet et m_l le nombre de paquets nécessaires à sa reconstruction, on a :

$$lp = \sum_{l=1}^{L} \frac{f_l}{m_l}.$$

La répartition des L flux dans N paquets est schématisé dans la figure 6.1.

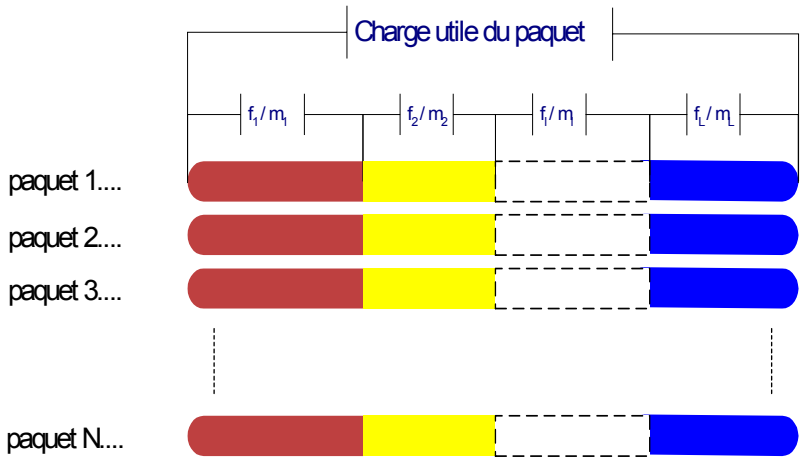


Fig. 6.1: Répartition des flux dans N paquets.

Algorithme de calcul de la fonction de priorité

début

red $\leftarrow$ *1*

*/*Tous les flux ont besoin des N paquets pour être décodés (pas de protection)*/*

fonction_de_priorite $\leftarrow$ *[N, N, N ...N]*

c $\leftarrow$ *densités de probabilités(N, p)*

charge_utile $\leftarrow$ *red*$\times$*taille_totale_des_flux / N*

Tant que top_qualite < qualite_requise ou red < cout_max faire

*/*top_ qualite est la qualité max que l'on peut obtenir à redondance donnée */*

Pour chaque flux s allant de 1 à L-1 faire

/ Si m_s = N + 1, le flux s est non transmis */*

Pour m_s allant de m_{s-1} à N + 1 faire

*/*calcul niveau de protection du flux L sachant l'occupation des flux d'indice inférieur à L et la charge utile de chaque paquet */*

$m_L \leftarrow$ charge_ utile - charge_ occupee

Si $m_L > m_{L-1}$ alors

qualite($m_1, m_2, ...m_L$) $\leftarrow$ increment_ qualite(1,2, ...L)$\times$c($m_1, m_2, ...m_L$)

*/*Calcul pour chaque L-uplets $\overline{Q}$ */*

FinSi

FinPour

FinPour

top_ qualite = max(qualite)

fonction_ de_ priorité $\leftarrow$ [$m_1, m_2, ...m_L$]

*charge_ utile $\leftarrow$ charge_ utile + 1 /*La taille de la charge utile est incrementée de 1 élément */*

*red $\leftarrow$ charge_ utile * N / taille_ totale_ des_flux /*La redondance se trouve modifier*/*

fin TantQue

Fin

6.2.2 À propos de la qualité

La description de la source concentre la moitié des efforts dans la réalisation d'un schéma de codage conjoint source-canal. On peut distinguer dans un ordre croissant de difficultés les cas d'une source construite par un seul schéma d'acquisition d'une source communément appelée multimédia.

Qualité pour un seul média

La source est composée dans ce cas d'un seul média (son, image, vidéo). L'estimation de la qualité est un sujet de recherche qui préoccupe de nombreux membres de la communauté image. La mesure classique qui s'impose encore aujourd'hui est le *Peak Signal to Noise Ratio (PSNR)* défini comme suit pour des échantillons sur 8 bits et par rapport à l'écart quadratique moyen (EQM)

$$PSNR = 10 \log_{10} \frac{255^2}{EQM}$$

avec

$$EQM = \frac{1}{N}\sum_{i=1}^{N}(x_i - \widehat{x}_i)^2.$$

Ce critère objectif est fortement remis en cause dans l'évaluation de la qualité particulièrement en image. Un PSNR identique peut être en effet obtenu aussi bien pour des dégradations localisées que réparties sur toute l'image. La figure 6.2 temoigne des différences perceptibles engendrées. La perte d'un bloc 32×32 dans l'image de gauche est préjudiciable à la qualité globale qui est conservée dans l'image de droite suite à des dégradations réparties sur toute l'image (codage SPIHT à 0,158 bpp).
Une meilleure évaluation de la qualité est réalisée directement par des observateurs humains dans un environnement normalisé [ITU 00]. Le coût des tests subjectifs motive cependant la création de nouveaux critères en bonne correspondance avec le jugement humain. Dans cette voie, des critères sont proposées spécialement pour le codeur JPEG [Watson 93] ou indépendemment du codage source [LeCallet 01].

Fig. 6.2: Images Léna dégradées comportant un même PSNR de 32,09 dB. L'image de gauche comprend une dégradation très localisée alors que l'image de droite est issue d'un codeur SPIHT (0,158 bpp).

Qualité multimedia

Le problème de l'évaluation de la qualité se revèle plus complexe pour une source multimédia. Les niveaux de priorité d'une source composée de textes, de sons, d'images et de vidéos (*e.g.* compte rendu médical multimédia) sont dans ce cas en forte dépendance avec l'application. Pour une source audiovisuelle, une contribution récente est formulée au

sein de l'ITU[ITU 01]. Elle fait état de l'importance de la vidéo dans notre perception par rapport au son. La corrélation entre la qualité vidéo et la qualité globale audio visuelle est plus grande que la corrélation entre la qualité audio et la qualité globale audiovisuelle. Un modèle est en outre proposé en bonne correspondance avec les données subjectives. Ce travail préliminaire précise néanmoins la portée limitée du modèle à certaines scènes audio visuelles.

Mesures de qualité : application à la transmission d'images

Fig. 6.3: Décodage du flux 1 et reconstruction de l'image bateau 512×512 (débit 0,4 bits/pixels pour un PSNR de 32,22 dB).

On propose de transmettre l'image Bateau de dimension 512×512 sur un canal dont le profil de pertes répond à une loi exponentielle. Un codage de source de type SPIHT est tout d'abord réalisé sur l'image (débit de 0,8 bits/pixel). Pour une première application

optimisée, nous diviserons simplement le flux source en deux sous-flux binaires. Le flux 1 est composé de 104000 bits et délivre seul au décodage l'image de la figure 6.3. Le flux 2 est composé de 105720 bits et délivre en couche de raffinement en complément du flux 1, l'image de la figure 6.4.

Fig. 6.4: Décodage du flux 2 et reconstruction de l'image bateau 512×512 (débit 0,8 bits/pixels pour un PSNR de 37,02).

La dispersion de la source est effectuée dans 32 descriptions équivalentes. Compte tenu du volume total d'information (209720 bits) et si l'on suppose un lien IP sur Ethernet, la taille des paquets reste en deçà de la fragmentation avec ce partage. La recherche du nombre de projections nécessaires pour la reconstruction de chaque flux est réalisée avec l'algorithme présenté au § 6.2.1 pour maximiser l'espérance mathématique de la qualité. Soient m_1 et m_2 les nombres de projections nécessaires pour la reconstruction respectivement des flux 1 et 2. La contrainte qu'exerce la capacité du canal est pour

l'instant relachée. Le taux de redondance limite n'est pas spécifié. La recherche est donc exhaustive et n'est contrainte que par la seule satisfaction de la qualité exigée (inférieure à la mesure de qualité lorsque les flux 1 et 2 sont correctement reçus). La figure 6.6 illustre le résultat de cette recherche en représentant pour un couple (m_1, m_2), l'espérance mathématique de la qualité correspondante et pour une valeur moyenne de perte de 10 %. On remarquera figure 6.6 la contrainte sur les niveaux de protection à savoir $m_2 > m_1$.

Fig. 6.5: Image requise par le service. L'acceptabilité de service se définit ici simplement par un niveau de qualité de 34,25 dB (0,55 bits/pixel).

Les valeurs entières sur lequel est défini le nombre de projections limitent le cardinal de l'ensemble des solutions à 393 couples possibles. Pour un seuil de qualité maximale correspondant à la contrainte de qualité minimale soit dans l'exemple égal à 37 dB, on obtient $m_1 = 7$ et $m_2 = 11$ pour un taux de perte moyen[31] de 10 %.Ainsi sept projections

[31] La loi exponentielle n'est pas spécialement une distribution discrète. Pour un nombre de paquets faible,

suffisent à reconstruire le flux 1 (image correspondante figure 6.3) et onze projections suffisent à reconstruire le flux 1 et 2 (image correspondante figure 6.3). Si une version dégradée intermédiaire entre la réception du flux 1 et du flux 2 est requise, l'intersection entre la courbe de la figure 6.6 et le seuil demandé fournit des niveaux de protection moins stricts.En prenant l'exemple de l'image Bateau de la figure 6.5 comme image acceptable pour le service supposé, le seuil placé à 34,25 dB nous donne $m_1 = 23$ et $m_2 = 26$. Le résultat de l'intersection du seuil et de la courbe de l'espérance mathématique de la qualité est donné figure 6.7.

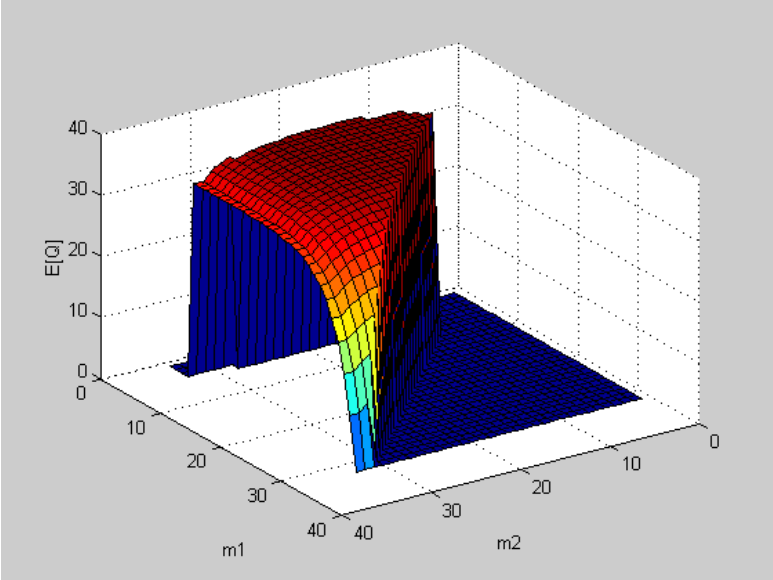


Fig. 6.6: Calcul de l'espérance mathématique de la qualité pour les couples $(m1, m2)$ et pour un profil de perte de paquet exponentiel de valeur moyenne 10 %.

elle présente une différence entre la valeur moyenne spécifiée et le calcul direct du moment d'ordre 1. Dans l'exemple de 32 paquets avec une valeur moyenne de perte à 10 %, l'erreur est d'environ 15 %. La valeur moyenne calculée est légèrement plus faible. Cette erreur diminue cependant fortement pour un plus grand nombre de paquets.

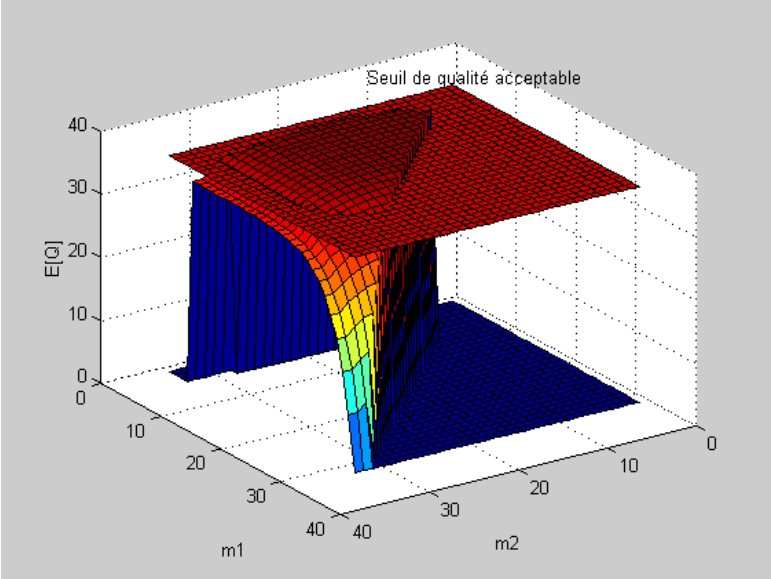


Fig. 6.7: Résultat de l'intersection du seuil de qualité acceptable avec la courbe d'espérance mathématique de la qualité.

Les figures 6.8 à 6.11 illustrent le comportement de l'optimisation en fonction du taux de perte de paquet moyen et du niveau de qualité exigée à la réception. L'ensemble de ces résultats est réalisé à partir de la même source que précédemment à savoir l'image Bateau répartie en deux sous-flux (1 et 2) à la sortie d'un codeur SPIHT. Ces deux flux sont ensuite codés dans 32 descriptions équivalentes (nombre de paquet fixée et longueur de paquet variable). Pour chaque valeur moyenne de perte, on indique les couples (m_1, m_2) correspondants aux nombres de projections nécessaires pour la reconstruction respectivement des flux 1 et 2. Les quatre figures (6.8, 6.9, 6.10, 6.11) coïncident à quatre niveaux décroissants de qualité espérée à la réception.
La figure 6.8 indique les valeurs de protection m_1 et m_2 pour un niveau de qualité espérée à la réception de 36 dB. La qualité maximale (réception des flux 1 et 2) étant de 37,02 dB et la réception du seul flux 1 fournissant une qualité de 32,22 dB, les deux flux sont sollicités pour le décodage. On assiste ainsi à une protection relativement proche pour les deux flux avec néanmoins $m_1 \leqslant m_2$. Les deux fonctions sont alors décroissantes correspondant à une protection accrue du fait de l'augmentation du taux de perte. Cette décroissance disparaît cependant pour m_2 avec un taux de perte élevé (35 %) : m_1 reste décroissant et m_2 occupe la place qui lui reste. Une faible variation du taux de perte tend à modifier la répartition des niveaux de protection. L'optimisation entre 35 % et 40 % de perte fait état d'un premier plateau pour m_2 et d'un changement de pente pour m_1. La décision est de préserver le flux prioritaire (flux 1) par rapport à un même niveau de protection pour le flux de raffinement (flux 2). La décision inverse est prise par l'algorithme pour un taux de perte légèrement supérieur (40 % et 45 %). La dernière tendance est de miser sur la pro-

tection du flux 1 par rapport au flux 2 pour un taux de 50 % de perte. Ce comportement est explicable par l'exigence en qualité de réception comparée à ce que peut réellement fournir la source et ce dans un environnement à fort taux de perte.
La figure 6.9 indique les valeurs de protection m_1 et m_2 pour un niveau de qualité exigée à la réception de 34 dB. Ce niveau de qualité est plus en accord avec la source considérée. Le comportement de l'algorithme peut se résumer par un nombre de projections nécessaires monotone décroissant pour m_1, pour reconstruire le flux prioritaire, et par une sollicitation variable pour le flux 2, le flux de raffinement. Le flux 2 est effectivement très sollicité pour des taux de pertes faible (de 5 à 20 %). Pour des taux de pertes intermédiaires, le flux 1 bénéficie d'une protection accrue au détriment du flux 2 (plateau à 20 %). Enfin, la nécessité du flux 2 est pressentie pour des taux de pertes plus important (30 %) afin d'obtenir la qualité de 34 dB.
Les figure 6.10 et 6.11 indiquent les valeurs de protection m_1 et m_2 pour deux niveaux de qualité respectivement de 32 dB et 29 dB. Dans ces deux cas, la réception du seul flux 1 est acceptable compte tenu de son niveau de qualité à 32,22 dB. Dès lors, les différences entre les niveaux m_1 et m_2 sont d'autant plus significatives que le flux 2 devient superfétatoire (pouvant aller jusqu'à la décision de ne pas l'envoyer pour $m_2 = 33$).

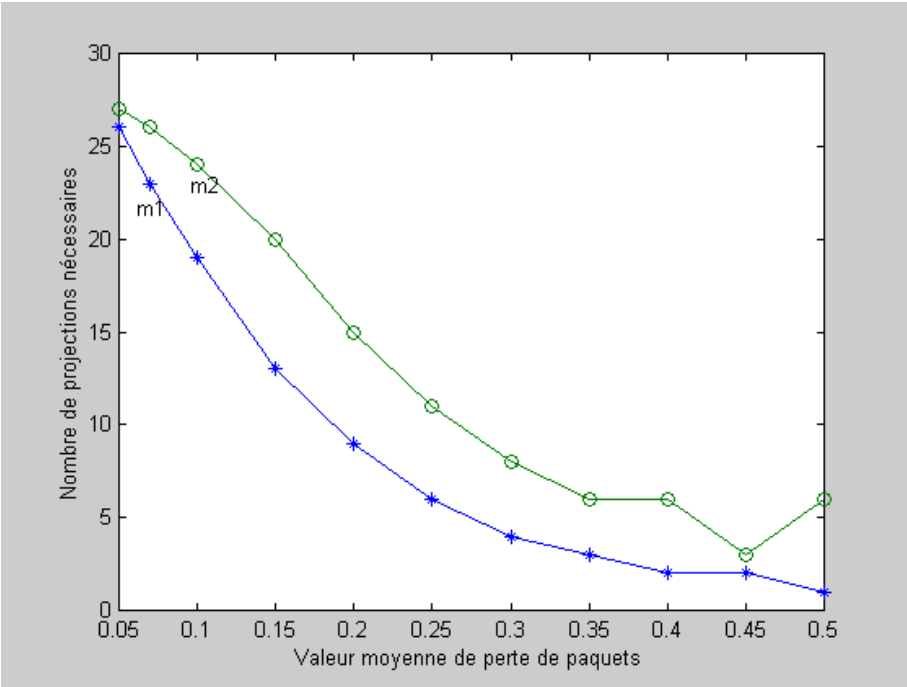


Fig. 6.8: Niveaux de protection m_1 et m_2 pour les deux sous-flux source (1 et 2) pour un niveau de qualité acceptable de 36 dB.

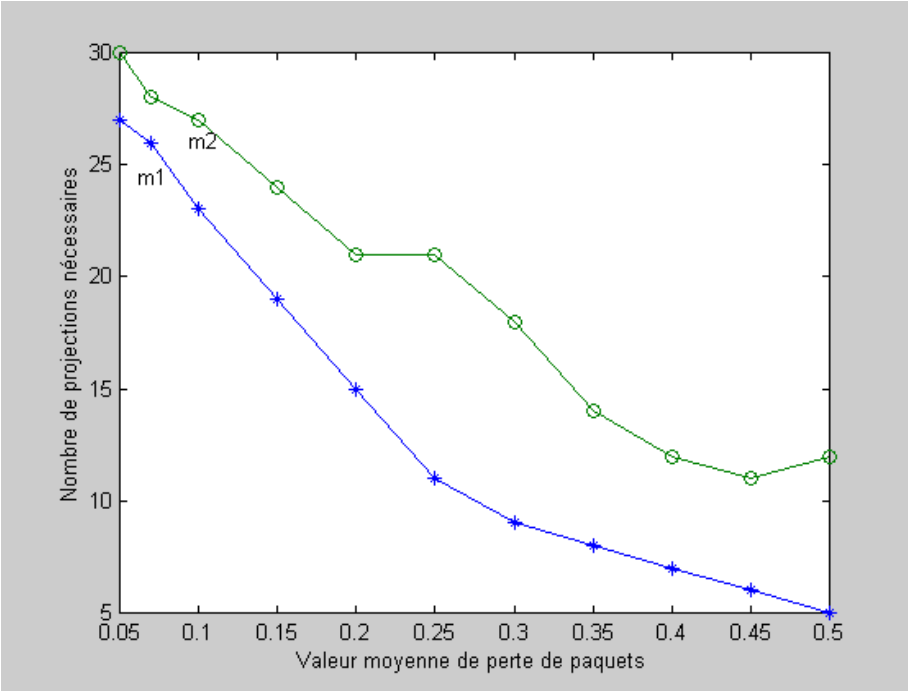


Fig. 6.9: Niveaux de protection m_1 et m_2 pour les deux sous-flux source (1 et 2) pour un niveau de qualité acceptable de 34 dB.

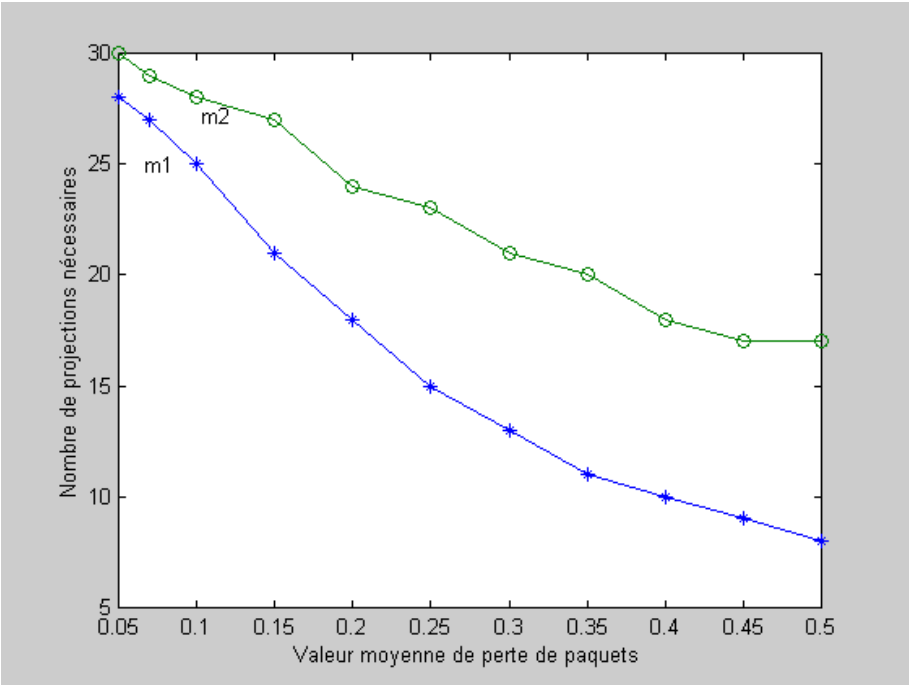


Fig. 6.10: Niveaux de protection m_1 et m_2 pour les deux sous-flux source (1 et 2) pour un niveau de qualité acceptable de 32 dB.

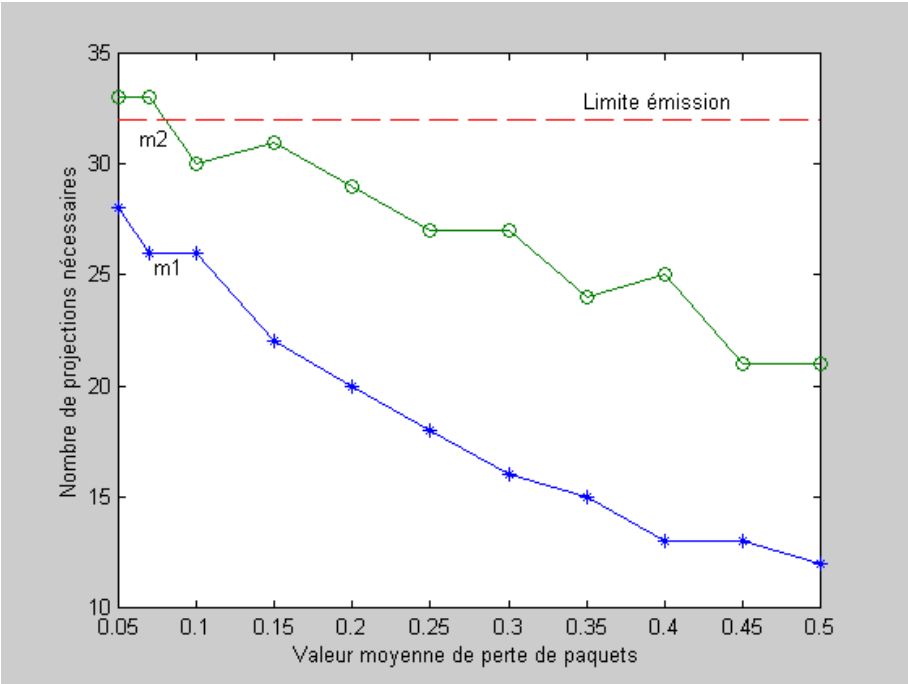


Fig. 6.11: Niveaux de protection m_1 et m_2 pour les deux sous-flux source (1 et 2) pour un niveau de qualité acceptable de 29 dB.

6.2.3 Voies de retour canal

La méthode d'estimation des niveaux de protection suppose en plus des incréments de qualité de la source une connaissance de l'état du canal. On expose une solution appliquée au cas réel de l'Internet. L'objectif est l'extraction de la probabilité de perte d'un flux.

RTP/RTCP

Mettre l'Internet dans une boîte noire n'est pas une chose facile. Bien que de nombreuses distributions (uniforme, binomiale, Zipf, Poisson, exponentielle...) soient avancées pour obtenir un modèle de canal, la caractérisation de l'Internet reste un sujet de recherche très ouvert [Paxson 97], [Savage 99]. Une alternative à la modélisation, pour notre calcul de l'espérance mathématique de la qualité est l'obtention de la probabilité des pertes réelles sur le lien emprunté. Outre l'horodatage et la remise dans l'ordre des paquets, le couple de protocoles RTP/RTCP [Schulzrinne 96] (*Real Time Protocol/ Real Time Control Protocol*) se singularise par l'évaluation des propriétés du canal. En effet, RTCP joue le rôle de voie de retour en effectuant les mesures du délais de transmission, de la gigue et du taux de pertes de paquets. Des rapports sont fournis à l'émetteur comme au récepteur (*Sender Report, Receiver Report*) sur les statistiques dynamiques du lien. Son indépendance avec le protocole de transport permet son utilisation aussi bien sur UDP que sur l'AAL5 d'ATM. Cependant ce protocole n'assure aucune fiabilité ni ne garantit un délai de livraison. Aussi,

l'association du codage prioritaire Mojette et de la voie de retour offerte par RTP/RTCP est proposée comme schéma de codage conjoint source/canal.

TCP-friendly

Les applications TCP font l'image de bons voisins à côté de celles qui ne mettent en œuvre aucun mécanisme qui évite la congestion du réseau [Jacobson 88]. L'effondrement du réseau est préjudiciable à l'ensemble des sources tout comme l'embouteillage en plein centre ville a pour conséquence de retarder l'ensemble des usagers. Quand une congestion est détectée (indiquée par la perte d'un ou plusieurs paquets), TCP réduit sa fenêtre de congestion par deux. Plus précisément, la perte de paquets influe sur la bande passante d'une connection TCP par la formule suivante [Madhavi 97] [Mathis 97]

$$BP = k\frac{MTU}{RTT \times \sqrt{p}} \tag{6.3}$$

où RTT est l'estimation dynamique du temps d'aller et retour (*Round Time Trip*), MTU l'unité de transport maximale de la connection (*Maximum Transport Unit*) et k une constante égale à 1,22. La probabilité de perte p est une estimation du transmetteur au lieu du taux de perte réel observé à la réception. La réactivité de l'application est la clé du succès de TCP sur l'Internet.
Cependant, TCP est extrêmement sensible à l'occupation de la bande passante par d'autres applications non TCP. Les effets se manifestent dans les variations du débit fourni par la source. En outre, TCP n'est pas utilisable dans une application temps réel. Les retransmissions induites par l'absence d'accusé de réception (ACK) ne sont pas acceptables compte tenu des contraintes de délais.
Une adaptation des sources UDP amène l'utilisation des accusés de réceptions pour estimer l'état de la connection et permet un lissage du débit délivré. Dans cette version *TCP-friendly*, l'arrivée de chaque paquet est suivie d'un envoi d'accusé de réception afin d'estimer le RTT et la probabilité de perte. La largeur de la fenêtre temporelle sur laquelle est effectuée l'estimation est sujet au compromis entre la stabilité de la bande passante et la validité de l'estimateur de probabilité de perte. La figure 6.12 extraite de [Tan 99] illustre les débits obtenus pour une source TCP et deux sources TCP-friendly. Les fluctuations de débit sont absentes dans le cas courtois et les deux sources mettant en œuvre ce protocole cohabitent par un partage équitable de la bande passante.

indice flux	1	2	3	4	5	6	7	8
Q_s (dB)	25,02	26,76	28,03	29,03	30,01	30,84	31,55	32,22
q_s (dB)	25,02	1,74	1,27	1	0,98	0,83	0,71	0,67
indice flux	9	10	11	12	13	14	15	16
Q_s (dB)	32,93	33,62	34,45	34,97	35,47	35,99	36,53	37,02
q_s (dB)	0,71	0,69	0,63	0,72	0,50	0,52	0,54	0,49

Tab. 6.1: Niveaux de qualité en fonction du nombre de sous flux reçus

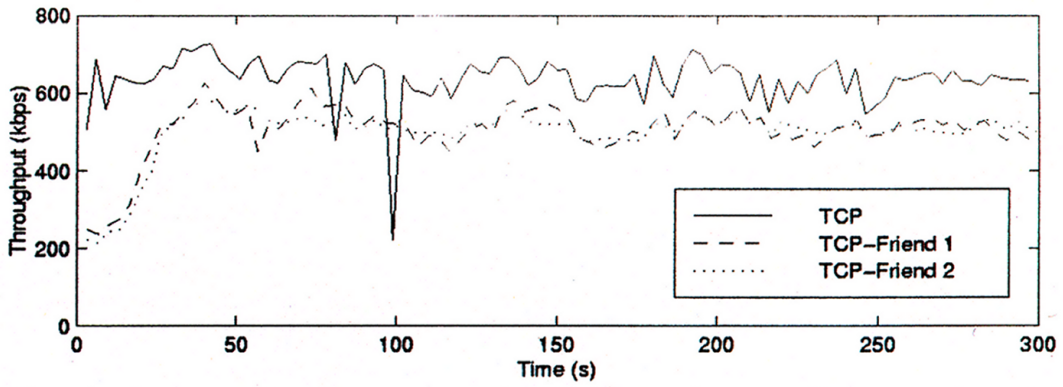


Fig. 6.12: Débit d'une source TCP et de deux sources TCP-friendly pour une connection entre Toronto et Berkeley [Tan 99].

6.3 Protection égale et protection inégale

Les résultats de cette section fournissent une comparaison entre les deux stratégies de protection : une protection égale (EEP) et une protection inégale (UEP) réalisée pour la transmission de l'image Bateau codée au préalable par l'algorithme SPIHT. Le flux source est sectionné en seize sous-flux de même taille. Les niveaux de qualité correspondants au nombre de sous-flux reçus sont indiqués dans le tableau 6.1 (la reconstruction du flux s implique la reconstruction du sous-flux précédent). L'étude porte sur des tailles de paquets fixes et pour deux modes de transmission à débit constant et variable.

6.3.1 Débit constant

La figure 6.13 illustre le niveau de qualité à la réception en fonction de l'arrivée des descriptions pour une protection égale et pour une protection inégale. On travaille ici pour un nombre constant de 32 descriptions. Le code UEP sur cette première figure n'a pas été optimisé. L'allocation des niveaux de protection est effectuée de sorte que la réception

indice flux	1	2	3	4	5	6	7	8
m_s	16	22	23	23	24	24	25	25
indice flux	9	10	11	12	13	14	15	16
m_s	25	25	25	25	25	25	25	26

Tab. 6.2: Niveaux de protection pour chaque sous-flux

d'une description se solde par la reconstruction d'un sous-flux. En conséquence, la taille des paquets générés est de 8884 bits si le premier et le dernier flux sont respectivement reconstruits par 17 et 32 descriptions (le dernier flux n'admet aucune perte). Partant de cette taille, l'optimisation de l'EEP délivre un niveau de protection de $m_{opt} = 22$ pour un profil de pertes exponentiel avec une valeur moyenne de 10 %. En dessous de ce nombre de descriptions, aucun élément de la source ne peut être reconstruit. La qualité correspondante est calculée à partir d'une image pour un niveau de gris uniforme. À l'arrivée des 22 descriptions, la totalité des éléments source protégés est reconstruite pour une qualité de 36,49 dB. À ce moment de la réception, celle-ci dépasse fortement de 5,65 dB la qualité reçue pour la protection inégale non optimisée. L'arrivée de descriptions supplémentaires n'améliore pas la reconstruction de la source pour l'EEP. Mais, pour un niveau moyen de perte de 10 %, l'EEP apporte un meilleur contrôle d'intégrité. Pour une forte variation de la probabilité de perte estimé (phénomène observé classiquement sur un canal IP), l'UEP peut s'avérer plus performante (à taux de pertes élevé ou à taux de pertes nul). De plus, cette dernière méthode de protection respecte le codage de source par une reconstruction progressive de ses éléments à chaque arrivée de description.

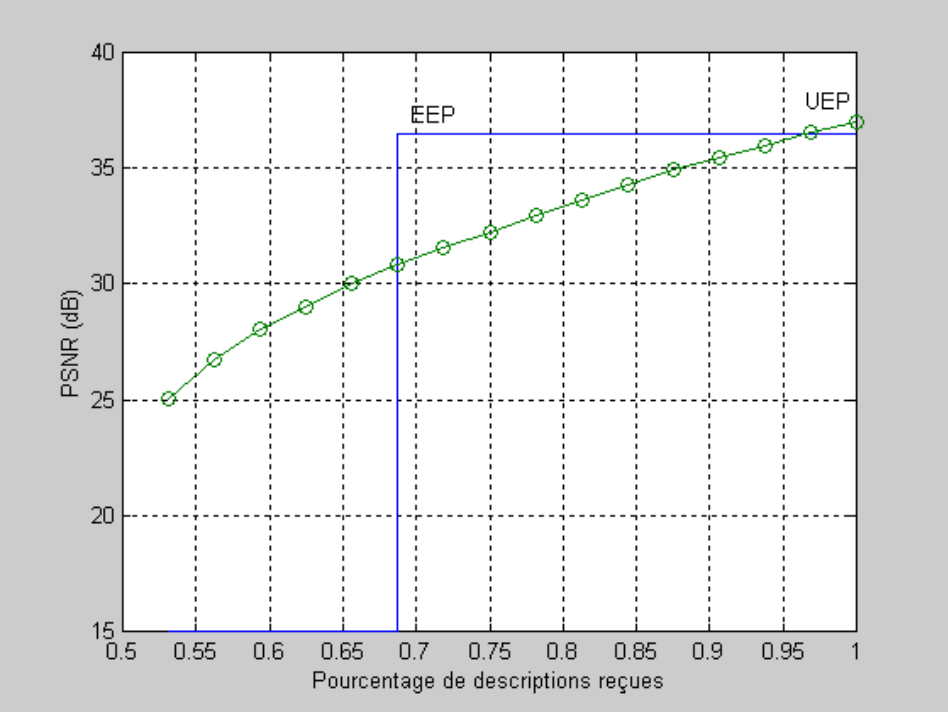


Fig. 6.13: Mesure de la qualité en fonction du nombre de descriptions reçues pour une protection égale (EEP) et pour une protection inégale (UEP) non optimisée.

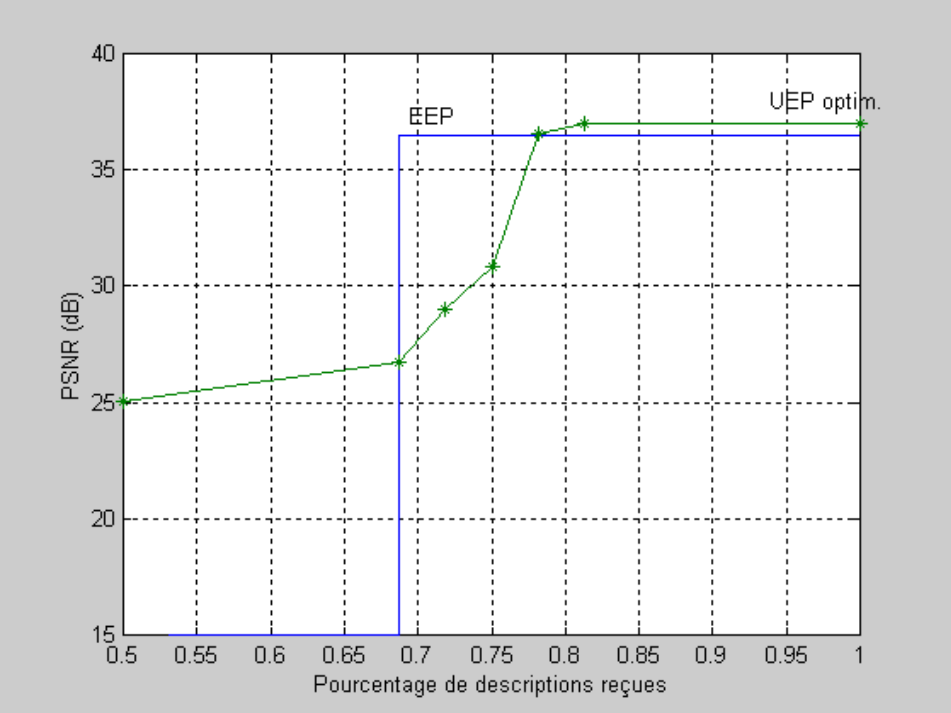


Fig. 6.14: Mesure de la qualité en fonction du nombre de descriptions reçues pour une protection égale (EEP) et pour la protection inégale (UEP) optimisée.

L'allocation optimisée est représentée figure 6.14. La stratégie employée fait état d'une agrégation des sous-flux. Cinq niveaux de protection distincts sont désormais sélectionnés. Le tableau 6.2 indique les valeurs de la fonction de priorité pour l'ensemble des 16 sous-flux. L'optimisation de la redondance permet de recouvrir un niveau de qualité supérieur dans le domaine de perte que pour la méthode EEP. Pour une variation de la probabilité de perte estimé, le schéma UEP demeure plus robuste. À 50 % de pertes (16 descriptions sont reçues), une version dégradée est obtenue.

6.3.2 Débit variable

Ce paragraphe présente les résultats dans le cas d'un débit variable disponible pour réaliser les méthodes de protection égales et inégales. L'application est de nouveau la transmission de l'image Bateau codée par l'algorithme SPIHT et décomposée en 16 sous-flux selon le tableau 6.1. Les figures 6.15 et 6.16 représentent la moyenne des qualités pondérées par la probabilité de réception pour les deux méthodes, pour une valeur moyenne de perte respectivement de 10 et 20 % en fonction du nombre de paquets envoyées. La qualité à la réception est augmentée dans le cas du codage UEP en regard du codage EEP pour les deux environnements de pertes. Les écarts vont de 0 dB (pour un nombre faible de descriptions) à 1,77 dB. Ils sont plus significatifs pour un taux de pertes de 20 %.

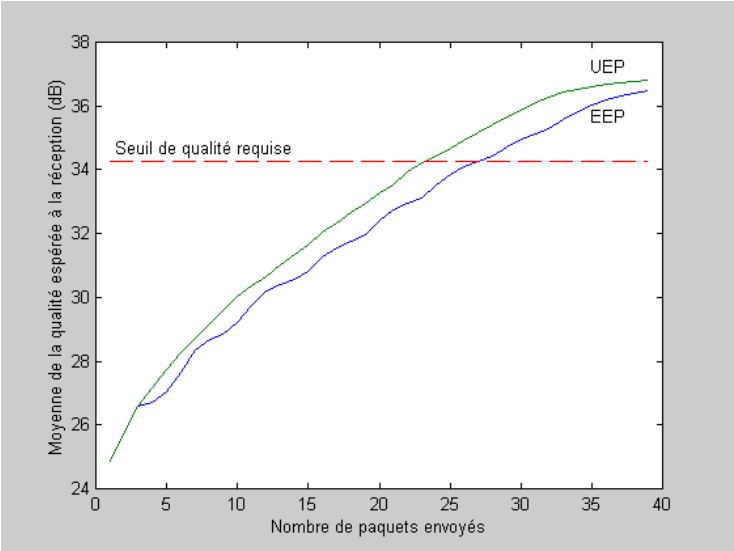


Fig. 6.15: Mesure de la qualité espérée à la réception en fonction du nombre de paquets envoyés pour un taux de perte moyen de 10 %. L'obtention de l'image de la figure 6.5 correspond au seuil de qualité (en pointillés sur la courbe).

D'autre part, pour une qualité requise à 34,25 dB (image de la figure 6.5), le nombre de paquets nécessaires est de 24 pour le code UEP et de 28 pour le code EEP. L'intersection du seuil de qualité avec les deux méthodes de codage est représentée figure 6.15. Le codage

prioritaire permet dans ce cas une réduction de 15 % de la bande passante utile. Ce codage apporte, vis-à-vis du code EEP, une fonctionnalité supplémentaire qui est d'obtenir rapidement une version même dégradée de la source.

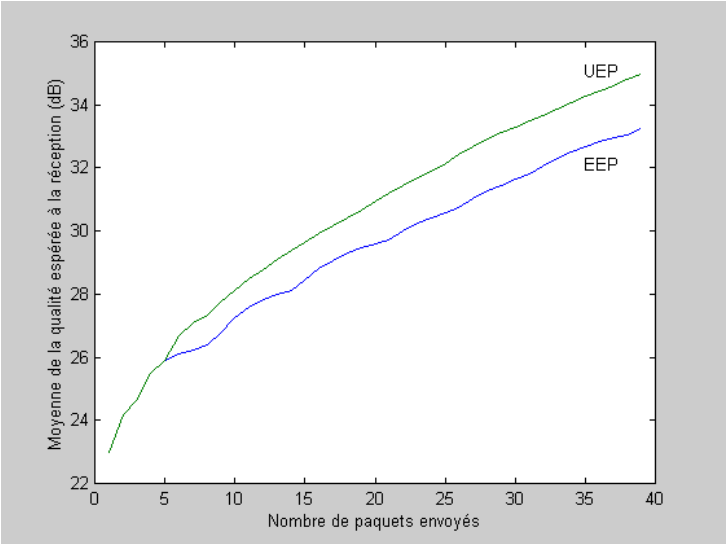


Fig. 6.16: Mesure de la qualité espérée à la réception en fonction du nombre de paquets envoyés pour un taux de perte moyen de 20 %.

6.3.3 Effet de pertes

La série d'images 128×128 (l'image originale est extraite de l'image test Clown) de la figure 6.17 illustre le déroulement lorsque que l'on effectue une requête sur une image avec deux procédés distincts : le transfert progressive SPIHT et le transfert avec un code à description multiple par transformation Mojette. Le premier transfert est combiné avec un protocole de retransmission mais pas le second.
En supposant que le logiciel d'affichage reconstruise une image au fur et à mesure que les paquets arrivent à chaque top d'horloge, l'affichage au cours du temps se déroule comme le décrit la figure 6.17.
Dans les deux cas la source est représentée dans six paquets dont le troisième est perdu pendant la transmission. Dans le premier procédé, la perte est temporaire puisque le paquet manquant est éventuellement retransmis et reçu au temps 20.

Jusqu'à la première perte le schéma progressif SPIHT est le plus performant. Cependant, la qualité de l'image est gelée en dépit de la réception des autres paquets tant que la retransmission n'a pas été complétée. En revanche, le code à description multiple a amélioré la qualité de l'image à chaque réception malgré la perte au temps 3. Bien avant la retransmission, le schéma à description multiple délivre l'image de meilleure qualité.

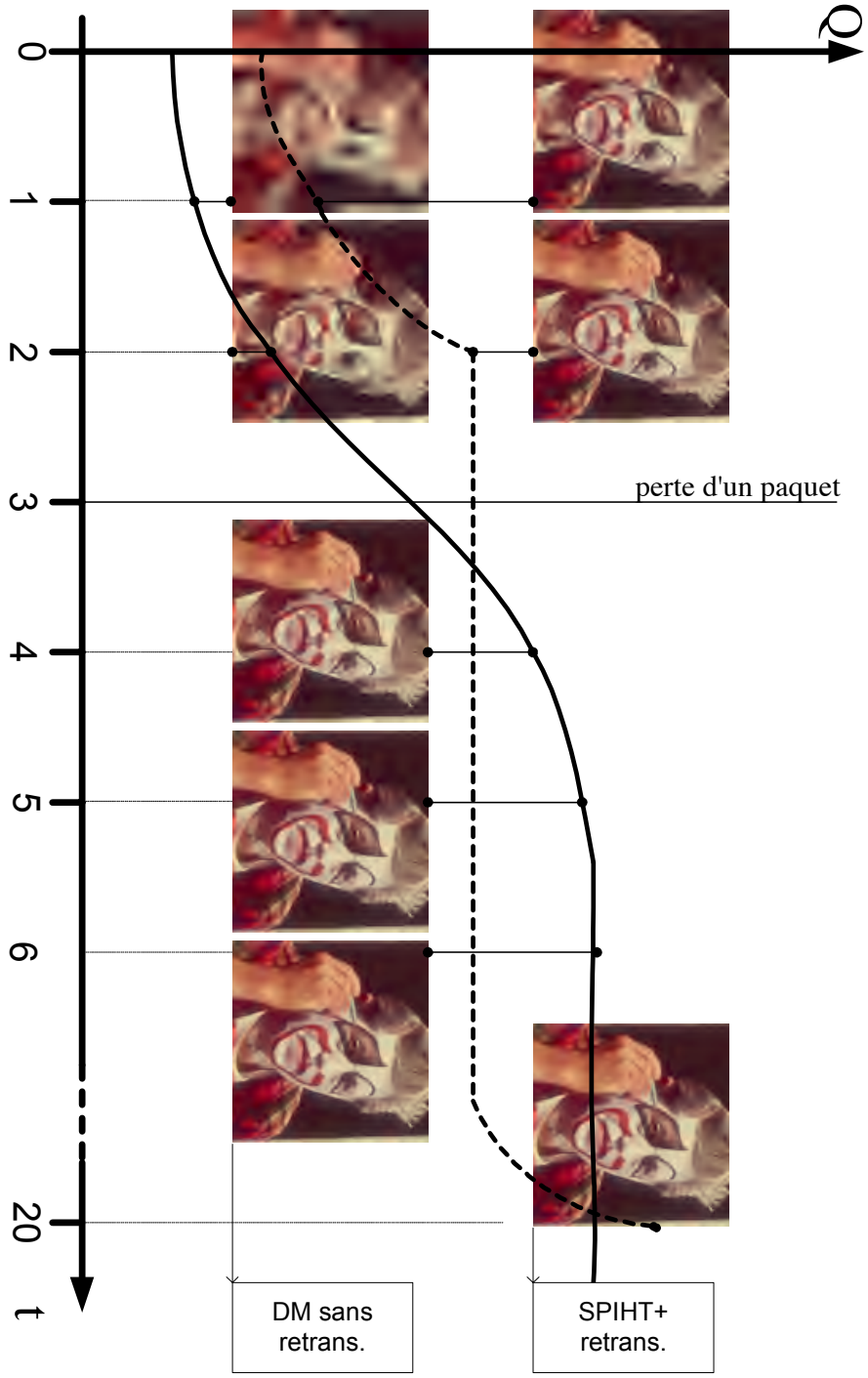


Fig. 6.17: Comparaison des reconstructions entre la description multiple (DM) et le codage SPIHT avec retransmission.

6.4 Conclusion

Dans une première section, nous avons mis en évidence les propriétés de description multiple de la protection inégale proposée au chapitre 5. La taille constante des descriptions et leur pouvoir de reconstruction équivalent (même quantité d'information reconstruite) confère au codage la propriété d'une description multiple équilibrée. De manière déterministe, un nombre donné de projections reçues (débit donné) induira un niveau de distorsion identique.
Nous avons apporté ensuite une solution pour optimiser le système de protection (*PET system*) en termes de redondance attribuée à chaque sous-flux source. Cette optimisation est contrainte conjointement par les propriétés de la source (incrément de qualité par sous-flux) et les caractéristiques du canal (probabilité de perte). Enfin, une comparaison est réalisée entre l'introduction de redondance pour une protection égale (EEP) et pour une protection inégale (UEP).

Les résultats de cette comparaison font état d'une augmentation de la qualité dans le cas d'une transmission avec codage UEP. Outre la fonctionnalité progressive conservée par le codage à description multiple, les qualités reçues sont supérieures au codage EEP. Pour un seuil de qualité requis (pour un service donné), les écarts de bande passante entre la protection inégale et la protection égale sont significatifs. L'optimisation du codage UEP focalisant le débit alloué sur les sous-flux prioritaires permet un gain en bande passante.

Conclusion de la deuxième partie

Le premier chapitre de cette deuxième partie nous a permis de construire au sens géométrique du terme le code $Moj(m, n)$. L'étape d'allocation des ixels sur un espace 2D correspond à la mise en mémoire tampon des éléments source avant leur transmission. La forme de la mémoire dans cet espace 2D est contrainte par la fonction de priorité demandée. La particularité de cette construction est de générer des projections de pouvoir de reconstruction égale. De ce fait, cette méthode de protection égale est parfaitement adaptée à la transmission sur des canaux à qualité de service non garantie. Une première comparaison avec les codes MDS, a été réalisée à l'issue du chapitre 4.
La protection simultanée de composantes hiérarchisées est mise en évidence par de nouvelles mémoires tampon "géométriques" qui supportent une reconstruction graduable à la réception. La composante prioritaire est reconstruite en premier avec un nombre de projections relativement faible et la composante dite de basse priorité nécessite un plus grand nombre de projections pour effectuer l'inversion de la transformation Mojette. La conception des supports d'information est de plus contrainte par la taille des projections. Le raffinement qu'apporte chaque projection dans la reconstitution de l'information originale témoigne des propriétés de description multiple que possèdent cette forme de protection graduable.
Dans le dernier chapitre, nous avons apporté une optimisation dans l'affectation du nombre de projections suffisantes pour la reconstruction de chaque composante. Le comportement pour une source délivrant deux sous-flux est illustré au travers d'environnements peu et fortement bruité (jusqu'à 50 % de perte). La protection est privilégiée naturellement pour le flux prioritaire. Dans le cas d'une requête qui nécessite l'ensemble des deux sous-flux, les niveaux de protection observent un durcissement comparable lorsque la probabilité de perte augmente. Une requête avec un niveau d'acceptation plus faible peut impliquer la décision de ne pas transmettre le flux. La qualité de service introduite dans cet exemple à deux sous-flux révèle l'adaptation de l'application aux contraintes du canal, de la source et du service demandé.
Des solutions réelles de voies de retour sont mises en avant pour informer notre codeur de l'état du canal. Le protocole de contrôle RTCP et le mécanisme TCP-friendly permet ainsi de fournir une estimation de la perte de paquets sur le lien emprunté.
Enfin, une comparaison avec le schéma de protection égale valide l'intérêt de la descrip-

tion multiple de l'information. Les qualités reçues à débits constant et variable témoignent d'un accroissement de la qualité dans le cas de la protection inégale (UEP). Ce gain de qualité s'accompagne d'une fonctionnalité de progression de la reconstruction la source, qui n'est pas incluse dans le cas de la protection égale (reconstruction "tout ou rien"). On peut néanmoins remettre en cause la validité du critère de qualité. Le PSNR est en effet un critère certes objectif mais qui ne prend pas en considération les aspects de perception du signal image. La réception d'une image d'un niveau de gris uniforme peut donner ainsi une valeur de PSNR équivalente à une image où la forme des objets constituants la scène est perceptible. Dans la perspective d'une meilleure définition de l'acceptabilité de service multimédia, il convient d'utiliser d'autres critères pour caractériser le contenu de la source.

Troisième partie

Applications sur réseaux

Introduction de la troisième partie

Les applications de la description multiple peuvent être divisées en deux groupes dédiées respectivement à la transmission et au stockage distribué d'information. Les problèmes des systèmes de stockage rejoignent cependant les problèmes de transmission sur réseau par le transfert de données (même de disque à disque) qu'ils nécessitent. Il est relativement simple de rapprocher les paquets qui transitent sur les réseaux de communication aux descriptions partielles du codage à description multiple.

Les paquets de données sont perdus sur les réseaux de données pour quantité de raisons. Ce peut être par exemple le résultat d'une congestion au niveau d'un nœud intermédiaire rencontré pendant le parcours du paquet. Sur l'Internet, ce phénomène engendre une probabilité de perte qui varie largement en fonction des heures de la journée, des jours de la semaine et du routage effectué.

L'hétérogénéité de l'Internet se révèle dans ses artères de communication et dans les terminaux qui s'y connectent. Ainsi, des artères en fibres optiques côtoient les liens sans fil et le nombre de terminaux mobiles rejoint celui des ordinateurs de bureaux. Les changements de bande passante qui résultent de cette diversité donne lieu classiquement à des dépassements de capacité des mémoires tampon des points d'interconnexion et à une élimination sans distinction des unités de transport.

L'approche conventionnelle dans ce cas est la réémission des paquets non acquités. Un protocole entre l'émetteur et le récepteur stipule alors comment ce dernier a ou n'a pas reçu un paquet. Quand les pertes sont sporadiques, ces retransmissions semblent être la meilleure solution : les paquets sont envoyés une seule fois et une faible charge supplémentaire est associée en voie de retour du récepteur. Si ces pertes augmentent, l'efficacité des retransmissions est remise en cause par une surcharge et des voies montantes et des voies descendantes favorisant d'autant plus la congestion. TCP sur IP résoud le problème en réduisant fortement (par deux) le débit d'injection des paquets.

L'augmentation des voies de retour est encore plus critique dans une application de diffusion. L'émétteur est souvent incapable de répondre à l'implosion des voies de retour et donc de satisfaire chaque destinataire.

L'alternative est donc de prévenir des pertes dès la première émission. Dans ce cas, le volume d'information est autonome et peu subir une série de dégradations qui aura pour conséquence une diminution progressive de la qualité. Nous présentons dans ce contexte,

deux types d'applications :

- dans le domaine des transmissions multimédias au chapitre 7 ;
- dans le domaine du stockage de données au chapitre 8.

Chapitre 7

Représentation multiple en transmission multimédia

Sommaire

7.1 Transmission d'images 150

7.1.1 Calcul de la fonction de priorité et de la mémoire tampon géométrique 150

7.1.2 Simulation de transmission avec pertes 152

7.2 Protocole de transfert ionosphérique 154

7.2.1 Conditions de transmission 155

7.2.2 Protocole de transmission 156

7.3 Description multiple à contraintes temporelles 156

7.4 Conclusion 158

L'apport de la description multiple en transmission multimédia sur des réseaux à qualité de service non garantie est indéniable. Les descriptions générées sont équivalentes et assurent une reconstruction progressive d'une version dégradée de la source en cas de perte. Le volume d'information ainsi codée est autonome car il ne nécessite pas de retransmission. Cette méthode répond donc à la fois aux contraintes de temps réel et d'intégrité particulièrement négligées sur des canaux IP. En outre, la suppression des retransmissions établit un bilan positif en terme de redondance pour les codages de type FEC avec dépendance du média qu'ils protègent.

La représentation multiple est donc largement applicable à la transmission multimédia sur canaux à effacement de paquets. On peut citer dans ce contexte la description multiple audio [Arean 00] congrue à la communication de la voix sur IP (VoIP) ou à la délivrance d'un contenu musical à partir de serveurs distribués. Des solutions sont aussi données pour des médias particulièrement sensibles telle que la vidéo [Servetto 99b]. Le volume important des données imputé à ce type de média nous amène à introduire la notion de description multiple à contraintes temporelles.

La première section de ce chapitre rapporte les résultats de l'optimisation de la description multiple Mojette appliquée à la transmission d'images. La deuxième section fait état de la transmission de données multimédia sur un canal ionosphérique. Des éléments sont fournis dans la troisième section pour adapter la méthode à la transmission d'un volume important de données telles que la vidéo ou les images médicales.

7.1 Transmission d'images

Cette section est extraite de [Parrein 01b] et illustre la mise en œuvre de l'estimation des niveaux de protection contrairement à l'application présentée au § 5.4 (page 116). Un codage SPIHT [Said 96b] remplace le codage JPEG progressif de l'application précédente. Le volume d'information contient 9830 octets décomposé en quatre sous-flux. Il est réparti dans seize projections de la série :

$$S = \{(p, q);\ p = -7, -6...7, 8,\ q = 1\}.$$

L'image originale envoyée provient de l'acquisition en IRM abdominale, 256×256 à 16 bits par pixel (figure 7.1, négatif de l'image originale).

Fig. 7.1: IRM abdominale 256×256 en 16 bits par pixel.

7.1.1 Calcul de la fonction de priorité et de la mémoire tampon géométrique

Un descriptif des flux est donné dans le tableau 7.1. Il indique pour chaque flux une mesure de qualité (PSNR), l'incrément de qualité et la taille correspondante. Les niveaux de priorités sont calculés sous la contrainte d'une redondance globale inférieure à 25 %. Le

	Flux 0	Flux 1	Flux 2	Flux 3	Flux 4
PSNR (dB)	9,26	24,81	29,31	34,20	35,70
dPSNR (dB)	.	15,55	4,5	4,89	1,5
taille (octets)	0	1638	2458	4096	1638

Tab. 7.1: Descriptif des flux en taille et en incrément de qualité. Le flux 0 correspond à la réception d'aucune projection ; le PSNR est calculé à partir d'une image avec un niveau de gris uniforme (valeur moyenne de l'image)

modèle de pertes est une distribution exponentielle avec une valeur moyenne[32] de 10 % (figure 7.2).

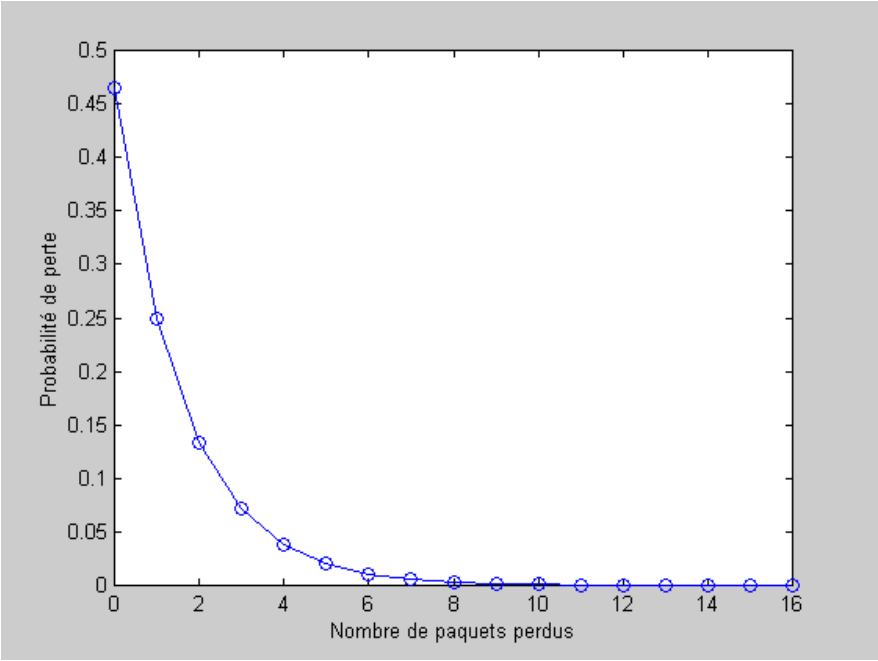


Fig. 7.2: Profil exponentiel de pertes pour une valeur moyenne de 10 %.

L'exécution de l'algorithme présenté au § 6.2.1 fait état de onze projections nécessaires pour reconstruire le flux 1, quatorze projections nécessaires pour le flux 2, quinze et seize projections pour les flux 3 et 4 respectivement. La mémoire tampon géométrique construite en conséquence, adaptée à cette fonction de priorité est représentée figure 7.3. La taille de chaque projection est de 786 bins.

[32] La loi exponentielle n'est pas spécialement une distribution discrète. Pour un nombre de paquets faible, elle présente une différence entre la valeur moyenne spécifiée et le calcul directe du moment d'ordre 1. Dans l'exemple de 16 paquets avec une valeur moyenne de perte à 10 %, l'erreur est d'environ 30 %. On perd en moyenne moins de paquets. Cette erreur diminue cependant fortement pour un plus grand nombre de paquets.

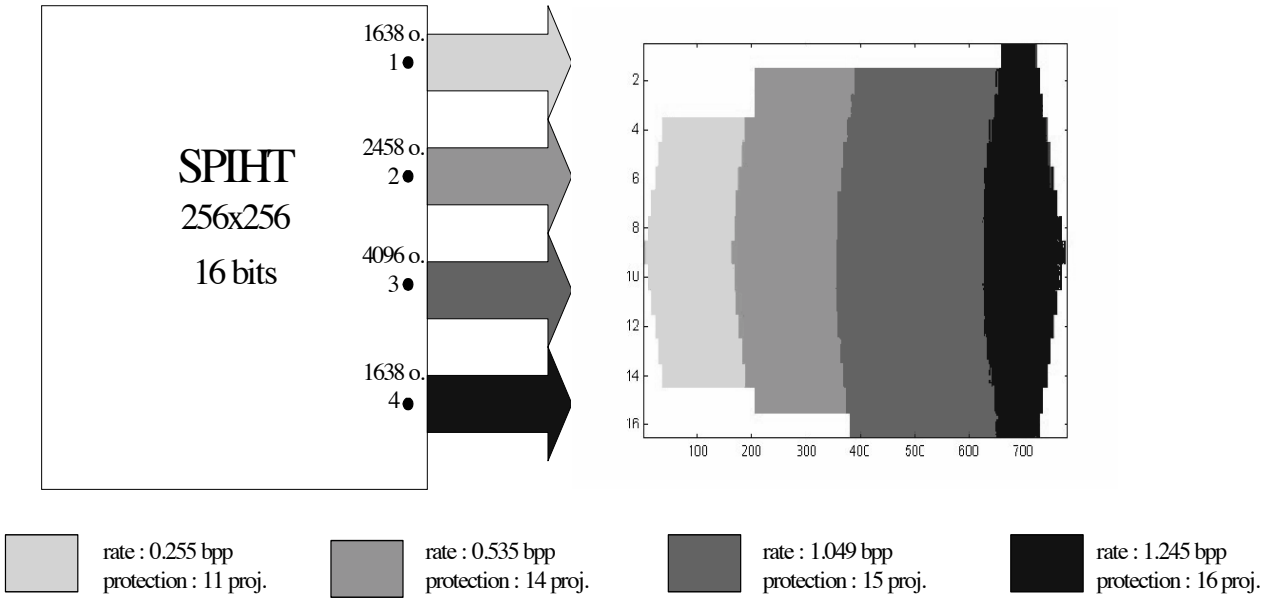


Fig. 7.3: Buffer géométrique construit pour la fonction de priorité indiquant que 11 projections sont nécessaires pour reconstruire le flux 1 d'une image SPIHT 256×256 ; 14, 15 et 16 pour les flux 2, 3, 4 respectivement.

7.1.2 Simulation de transmission avec pertes

L'image est reconstruite progressivement avec l'arrivée successive des projections. La figure 7.4 présente la mesure du PSNR en fonction du nombre de projections reçues. Une forte augmentation est décelable à chaque fois qu'un niveau de protection est atteint (#projections = 11, 14, 15, 16). En dehors de ces taux de réception, la qualité ne présente pas d'amélioration significative par la réception d'une projection supplémentaire. Néanmoins, la granularité de la source se définit au niveau des paquets reçus.

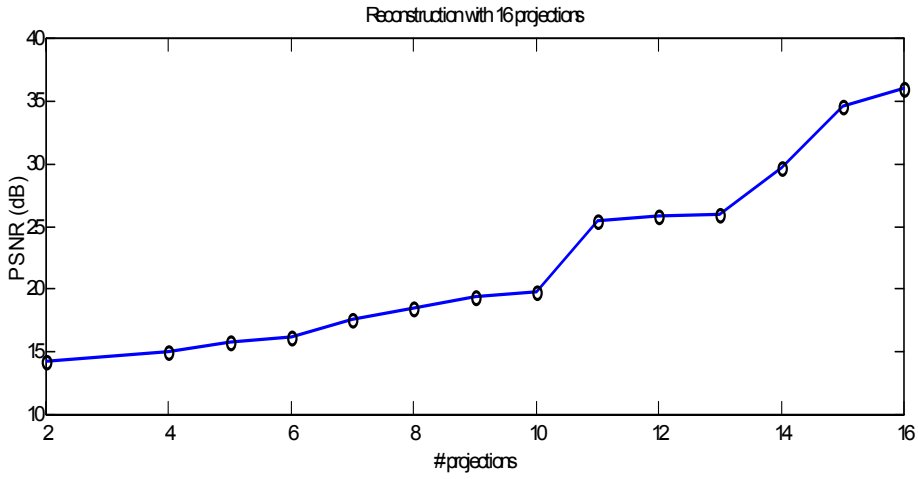


Fig. 7.4: Mesures du PSNR en fonction du nombre de projections reçues.

On présente sur la figure 7.5 les mesures de redondance effectuées à la réception soit le nombre de symboles reçus sur le nombre total de symboles message moins un. En dessous d'un certain nombre de projections (*i.e.* treize sur la figure), la redondance mesurée est négative. Le nombre d'éléments reçus est inférieur au nombre d'éléments qui composent la source. Cette zone d'utilisation permet une visualisation rapide sans occupation trop importante de la bande passante (*e.g.* vérification rapide dans une banque d'images médicales). L'augmentation linéaire de la redondance au delà de ce seuil de réception témoigne de la taille constante des projections. On remarque de plus que cette redondance est inférieure à 25 %. Les images reconstruites pour onze, quatorze, quinze et seize projections reçues sont affichées figure 7.6.

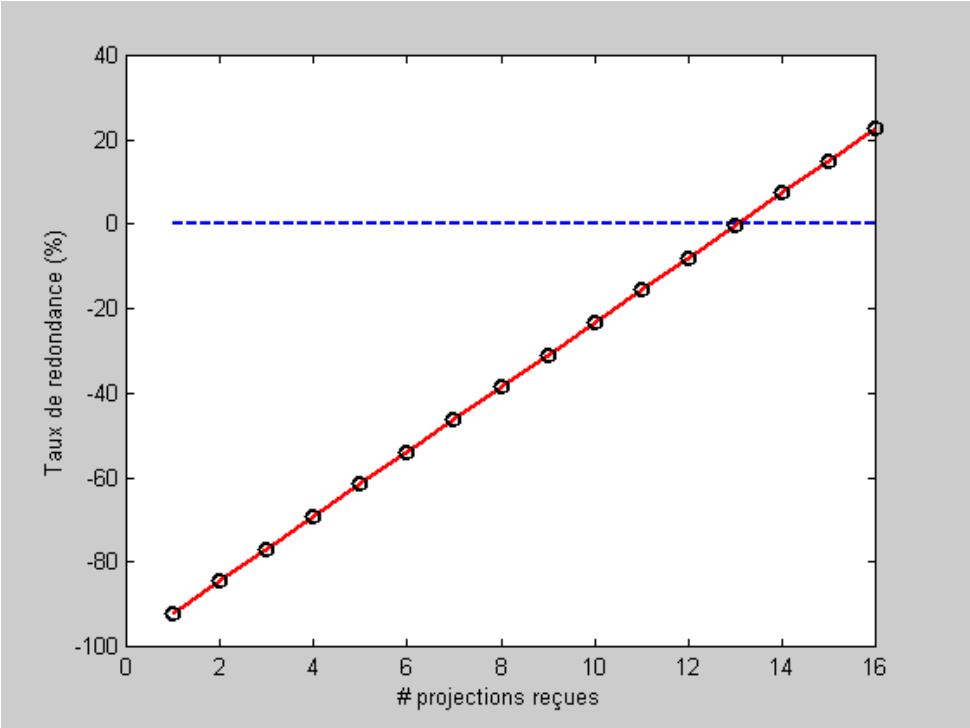


Fig. 7.5: Mesure de redondance à la réception en fonction du nombre de projections reçues.

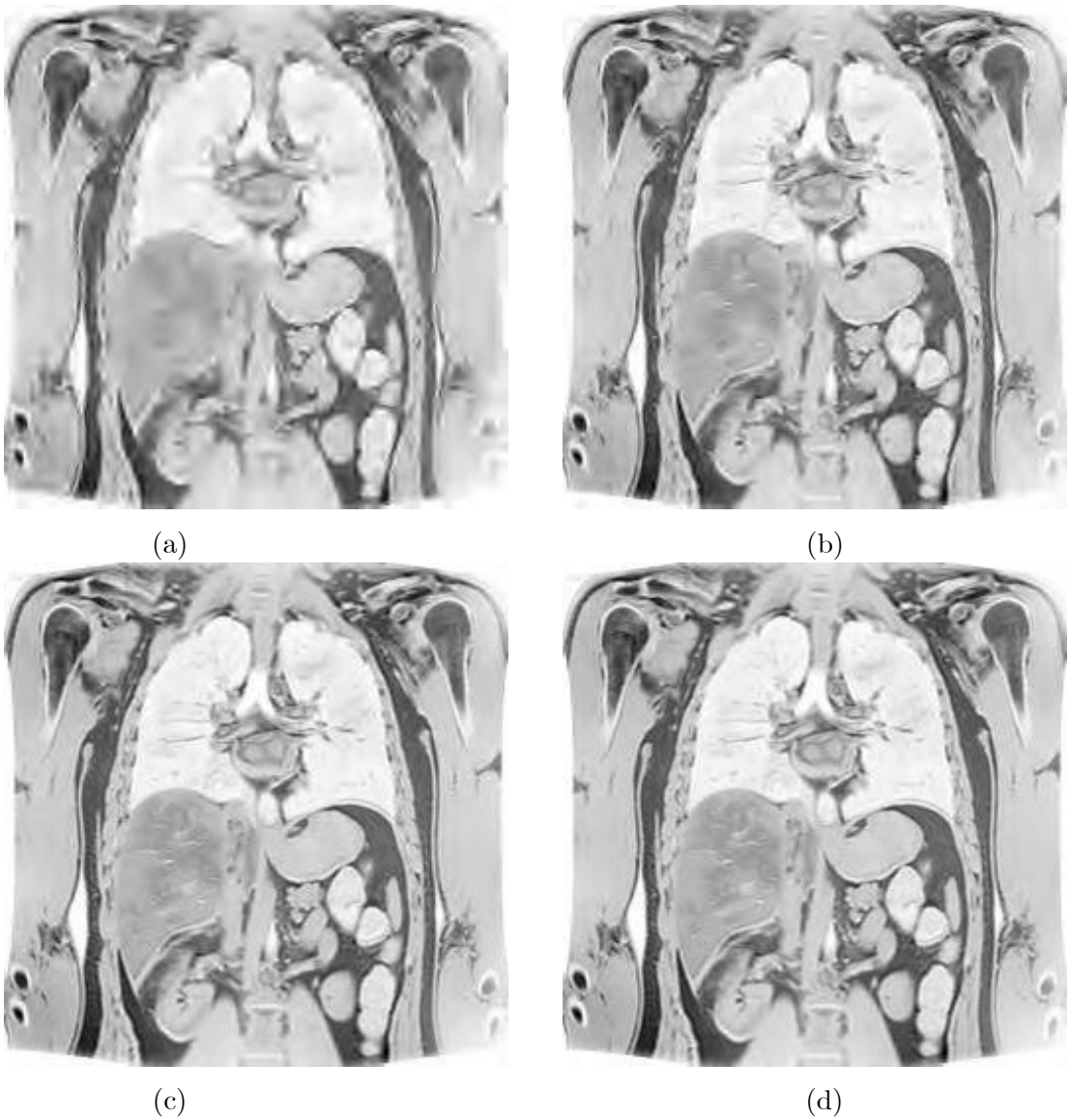


Fig. 7.6: IRM abdominales codés avec SPIHT. (a) débit = 0,255 bpp avec onze projections nécessaires. (b) débit = 0,535 bpp avec quatorze projections nécessaires. (c) débit = 1,049 bpp avec quinze projections nécessaires. (d) débit = 1,245 bpp avec seize projections nécessaires.

7.2 Protocole de transfert ionosphérique

Cette application s'inscrit dans le cadre de la collaboration avec le laboratoire IRCOM-SIC de Poitiers, sur le thème de la transmission de données au niveau de la ionosphère. Cette couche atmosphérique (entre 50 et 1000 km) est fortement sujet à des effacements de données et nécessite donc un contrôle d'intégrité tenant compte à la fois des propriétés

de la source et des caractéristiques du canal, particulièrement non stationnaires dans ce cas.

7.2.1 Conditions de transmission

Seules les ondes de Moyennes Fréquences (3-30 MHZ) ont la possibilité d'être réfractées dans la couche ionosphérique. Le milieu est du reste transparent et absorbant respectivement pour les ondes au-dessus et en-dessous de cette gamme de fréquences. Les réflexions successives sur la partie haute de l'ionosphère (couche F) et sur les couches plus basses (couche E ou sol terrestre) permettent une propagation sur des liaisons très longue distance. Par exemple, un trajet de longueur inférieure à 10 000 km ne nécessite pas plus de trois réflexions (figure 7.7). La distance maximale pour un bond est de 3500 km, limite physique et géométrique due à l'altitude des couches réfléchissantes et à la courbure de la terre [Escrig 97].

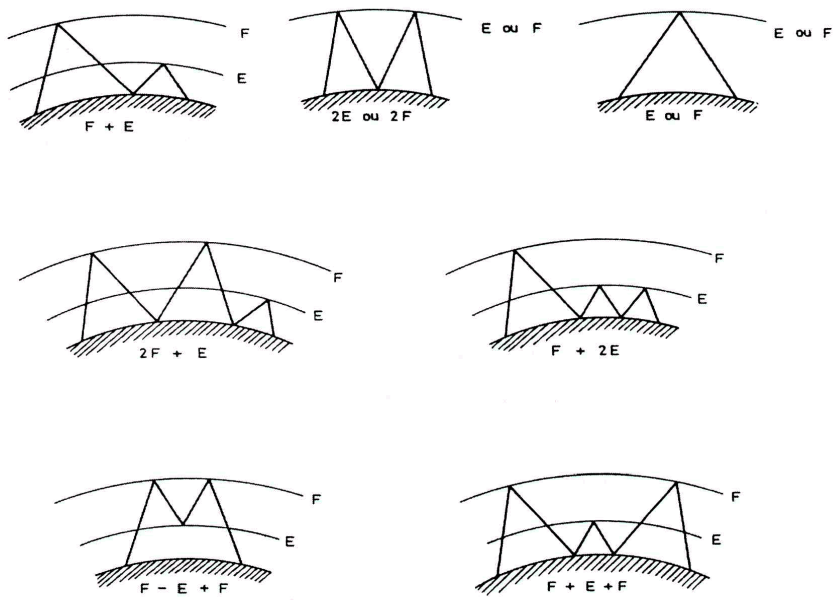


Fig. 7.7: Types de trajets ionosphériques possibles.

Les effets sur la propagation répondent aux phénomènes physiques. Nous ne citons ici que le phénomène d'évanouissement (fading), qui par des modifications du canal ionosphérique (changements de courbure des couches, rotation des axes de polarisation, variations de densité électronique), amène la perte du signal pendant une durée allant de quelques millisecondes à plusieurs dizaines de minutes. Cependant, l'expérimentation de transmission révèle un taux d'erreurs binaires relativement faible. L'exploitation optimisée du canal nécessite donc une compensation de pertes au niveau paquet adaptée aux conditions de

transmission.

7.2.2 Protocole de transmission

Le premier objectif est de récupérer au moins une version dégradée de la source. Dans ce cadre, l'utilisation de la représentation multiple est parfaitement adaptée pour que le contenu de la source soit résistante aux effacements et, dans le cas de perte, que l'on puisse tirer parti du maximum d'informations jusqu'aux limites contiguës au départ et à la fin de l'effacement. La segmentation du flux de données en petits paquets de taille fixe est de ce fait appréciable. Par une taille faible et constante, on entend :

- réduire le temps de remplissage des mémoires tampon en le rendant négligeable par rapport au temps de propagation ;
- faciliter le traitement des paquets en glissant vers le temps réel ;
- supprimer l'information de taille et de ce fait la protéger ;
- synchroniser et identifier les portions correctement reçues.

Il convient d'ajuster la taille des descriptions proportionnellement à la longueur moyenne d'un effacement et ce, de manière dynamique compte tenu des propriétés physiques non stationnaires du canal. On distingue alors trois cas d'effacement selon que sa longueur :

- correspond au temps de transmission d'un paquet et la perte ne peut être que constatée ;
- supprime une partie d'un paquet ;
- chevauche plusieurs paquets et entraîne la perte d'une partie de chacun.

Dans ces deux derniers cas, on peut exploiter la partie récupérée en cas de besoin. La reconstruction des mémoires tampon est en effet progressive au sein d'un même niveau de protection. On effectue la transformation Mojette inverse sur des "bouts" de projections, ceci réduisant la perte sur la longueur effective de l'effacement.
L'analyse des données reçues est similaire à la technique d'auto-cadrage utilisée en ATM. Un CRC est inclus pour délimiter et identifier les paquets. En **phase de recherche**, un CRC estimé est défini à partir d'un bit de donnée reçu et testé sur la série de bits qui le suit. La phase suivante de **présynchronisation** est atteinte par un décalage binaire bit à bit jusqu'à ce que le premier CRC valide soit localisé. Un nombre de paquets consécutifs doit être validé pour décider de **la synchronisation** du flux de donnée. Dès lors, l'algorithme à la réception n'aura de cesse de jouer avec ces trois phases en fonction du type d'effacement sur la liaison de transmission [Pakosz 01].

7.3 Description multiple à contraintes temporelles

Le problème de la description multiple devient plus complexe lorsque l'on veut transmettre sous cette forme un volume de données important telles que la vidéo. Un codage

appliquée sur la totalité des données n'est en effet envisageable qu'au prix du temps d'attente des données à coder. Des éléments de réponse sont cependant apportés par la création de mémoires tampon géométriques à trois dimensions (3-D) voire à n-dimensions (n-D) [Verbert 01].

Le temps de remplissage de mémoires tampon de très grande capacité est préjudiciable aux contraintes de temps réel même relativement souples. De ce fait, une solution hybride semble plus concevable. Des supports 2-D comme ceux présentés aux chapitre 4 et 5 peuvent être utilisés pour protéger des séquences de la source. Nous perdons alors la propriété d'équivalence de la description multiple puisque chaque description ne font pas référence à la même portion de données originales. Le système n'est alors plus déterministe dans sa globalité. Afin d'obtenir une grande probabilité de reconstruction, les supports 2-D peuvent être suivies de supports n-D où chaque ixel est multiplexé à la volée. Les projections de ces supports n-D contiennent dès lors des bins hautement corrélés ce qui peut être suffisant pour reconstruire les portions manquantes de la source.

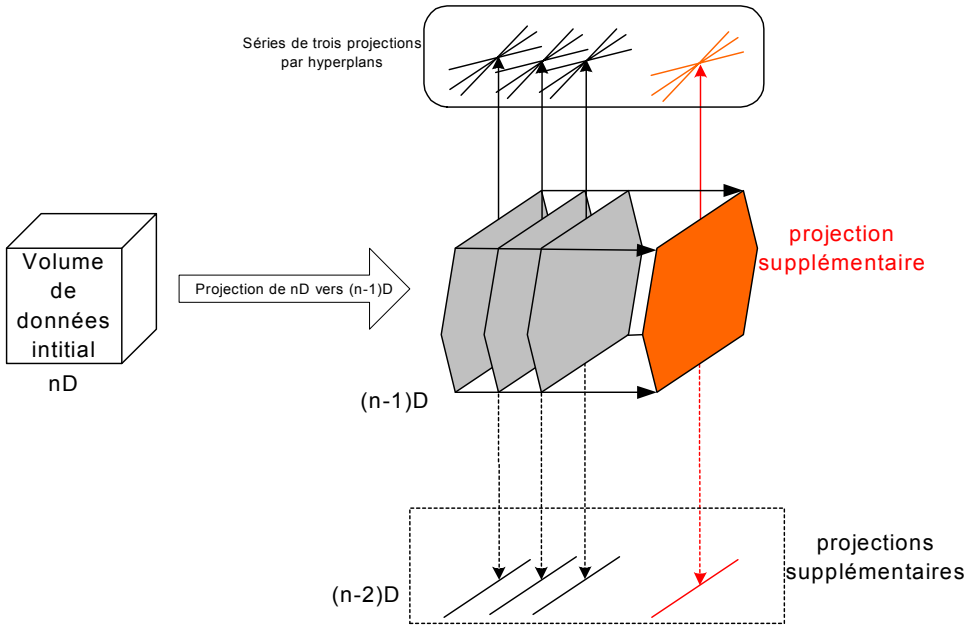


Fig. 7.8: Représentation de la solution hybride de protection combinant les projections (n-1)-D et (n-2)-D.

La figure 7.8 illustre la procédure de protection. Le volume de données initial est tout d'abord projeté selon l'algorithme Mojette par différents angles en un hyperplan de dimension n-1. Toute les projections sont réarrangées dans un autre hypervolume. Sur la figure, trois projections de dimension (n-1) sont calculées. Le deuxième hypervolume est composé de trois coupes (trois hyperplans de taille égale). Un hyperplan supplémentaire est calculé en corrélant toutes les coupes et en supposant un réarrangement de toutes les grilles des hyperplans. Malgré un partitionnement du volume de données à transmettre,

chaque hyperplan reste trop important en nombre de données pour être transmis tel quel. De plus, on tolère par ce système la perte de trois projections. Et, dans cette configuration la perte de deux projections (deux hyperplans) compromet la reconstruction de l'hypervolume initial. Aussi, les hyperplans sont projetés dans la direction (n-2) et ce dans quatre directions (trois suffisantes et une supplémentaire). Ce sont ces dernières projections qui sont transmises en mode paquet. Dans cette configuration, la perte de trois paquets ne compromet pas la reconstruction du volume initial de données.

7.4 Conclusion

Des exemples d'application de la description multiple Mojette ont été présentés dans ce chapitre. Le première exemple porte sur les résultats de l'optimisation de la description multiple appliquée à la transmission d'images. Le deuxième exemple illustre l'utilité d'un contrôle d'intégrité par la transformation Mojette sur un canal particulièrement bruité comme le canal ionosphérique. Des éléments sont fournis enfin pour adapter la méthode à la transmission d'un volume important de données comme de la vidéo ou encore des images médicales.

La diversité des applications exprime la portabilité de notre méthode de codage conjoint source-canal. En effet, le multiplexage des flux n'impose aucune distinction sur la profondeur en bits des éléments d'information. Les résultats du chapitre 4 indiquent cependant que le multiplexage binaire est préférable en termes de redondance introduite. Dans ces conditions toutes les sources multimédias peuvent être protégées comprenant les standards multimédias existants (JPEG-2000, MPEG-4). La méthode proposée n'impose pas en effet une restructuration des éléments aux niveaux du codage de source et donc ne freine pas l'évolution des sources actuelles. La protection est réalisée sur des éléments déjà quantifiés et hiérarchisés. Cette structure est fournie par l'application à notre codeur qui en accord avec les caractéristiques du canal (pertes, délais...) réalise la protection inégale idoine.

Chapitre 8

Représentation multiple en stockage de données

Sommaire

8.1 Système d'informations distribuées 160

8.1.1 Exemples de systèmes distribués multimédia sur IP 160

8.1.2 Rôle de la pile RTP/UDP/IP et des relations avec le système Mojette 161

8.1.3 Redondance projection/paquet 162

8.2 Application de télémédecine 163

8.3 Conclusion 164

Peut être plus que dans la communication par paquets, l'application de la description multiple dans le stockage distribué d'informations exprime toute son utilité. Considérons une base de données d'images stockées à différentes adresses électroniques via un codage à description multiple. L'utilisateur typique peut obtenir une copie locale de l'image ; et quand les besoins en termes de qualité augmentent, une ou plusieurs copies distantes peuvent être requises et combinées avec la copie locale pour rendre le service acceptable.
Classiquement, les bases de données résident sur des grappes de disques performants et sécurisés, où les données critiques sont dupliquées (technologies RAID). Les systèmes de stockage associés à ce modèle doivent garantir l'intégrité et la disponibilité des données précieuses. L'opération de *batch* consiste, dans un souci de sécuriser les données, à transférer pendant la nuit les travaux de la journée, à partir des disques magnétiques vers des bandes magnétiques. Le volume de données manipulées augmentant, la fenêtre nocturne devient critique, exigeant des vitesses de transfert à bande de plus en plus élevées.
Partant de ce modèle complètement centralisé, les architectures évoluent vers un modèle distribué en réseau, le stockage des informations étant de plus en plus sous traité par des entités externes au lieu d'utilisation. La spécialité des serveurs d'applications et l'intérêt économique (francs par giga-octet) contribuent en outre à cette répartition. Les systèmes

distribués comportent cependant des contraintes de temps réel et de sécurité plus fortes que leur pendant centralisé pour la reprise du contrôle de données éparpillées sur différents sites.
L'application exposée dans ce chapitre tient compte de ces aspects. La première section décrit un système d'informations distribuées sur IP en utilisant la transformation Mojette. La deuxième section précise le fonctionnement d'une architecture distribuée par transformation Mojette dans un contexte de télémédecine.

8.1 Système d'informations distribuées

La présentation de ce système est extraite de [Guédon 01a] où figure une architecture distribuée de serveurs d'images. Cette proposition détaille les avantages d'un environnement partagé en termes de répartition des charges réseaux, sécurité, dégradation progressive et facilité de synchronisation.

8.1.1 Exemples de systèmes distribués multimédia sur IP

La première catégorie d'exemples provient de données images archivées sur différents sites. La production d'image médicale représente un avant-goût critique. Les données patient sont disséminées entre différents hôpitaux nécessitant une description propre (format DICOM[33]). Par ailleurs, les fonctionnalités cryptographiques tant au stockage qu'à la transmission et de temps réels sont indispensables pour répondre à la demande du médecin consultant. Les petits hôpitaux dépassent aujourd'hui la production du Tbit/an et l'archivage de ces informations devient une priorité de l'**hôpital sans film**. Le problème de cette télémedecine depuis les cinq dernières années est de construire une infrastructure d'images capable de gérer ces informations en temps-réel légèrement différé (entre une minute et une heure pour l'obtenir). Même si cet intervalle temporel paraît large, le volume des données images pour certain examen (IRM fonctionnelle) est de l'ordre du Gbit. Sa transmission sur un seul canal IP est vouée à une congestion systématique.

D'autre systèmes distribués multimédia sont importants. La deuxième catégorie est la production d'images en ligne lors d'une session de téléconférence. Dans ce cas, chaque participant doit recevoir les données image de tous les points connectés. Le téléenseignement et le téléstaff en médecine peut être considérer comme un mélange des deux catégories précédentes puisque une partie de l'information est construite hors ligne mais peut représenter néanmoins une quantité importante de données. Le reste s'apparente aux procédé de diffusion associé à la vidéo conférence.

[33] Digital Imaging and Communications in Medicine

8.1.2 Rôle de la pile RTP/UDP/IP et des relations avec le système Mojette

Ces trois catégories de bases de données multimédia fonctionneront sur les réseaux IP dès lors que la pile classique TCP/IP ne monopolisera plus les échanges réseaux. En effet, l'intégrité de données assuré par TCP n'est pas compatible avec les contraintes temps réel de ces types d'applications. Aussi, un engouement pour UDP[34], jusqu'alors enfant pauvre des protocoles de transport, se manifeste actuellement lorsqu'il est utilisé avec le *Real-Time Protocol* (RTP). Ce dernier permet d'estampiller chaque paquet IP à des fins de synchronisation intra ou inter flux au niveau de l'équipement terminal multimédia. L'avantage pour la distribution vidéo est certain puisque même si une portion de l'image est dégradée, le flux de données se poursuivra. Cependant dans le cas de contrainte d'intégrité, la seule façon d'opérer demeure une version conjointe du codage source-canal pour prévenir ces dégradations en effectuant un mélange de fonctions temps-réel (maintenues par RTP) et de fonctions d'intégrité (assurées par la transformation Mojette). Ces deux fonctionnalités sont antagonistes dans la pile TCP/IP.

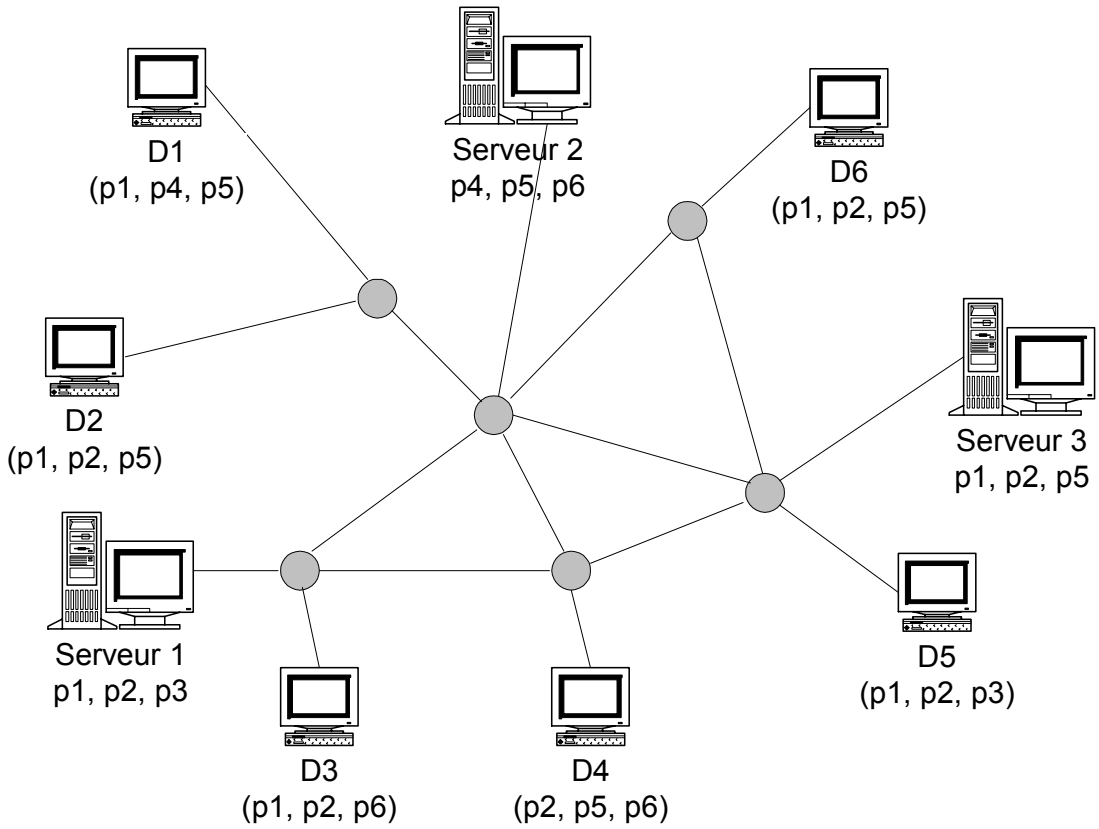


Fig. 8.1: Système distribué Mojette composé de trois serveurs et six clients.

[34] User Datagram Protocol

8.1.3 Redondance projection/paquet

Dans la suite, les données image sont réparties par la transformation Mojette en un nombre N de projections. La base de données est distribuée sur N sites distants. Lorsque un nombre donné d'utilisateurs effectue une requête sur cette base de données, le réseau IP doit résoudre une nouvelle distribution multipoint multipoint. Par les propriétés des projections exprimées aux chapitres 4 et 5, seul un sous-ensemble de M projections parmi N est nécessaire pour reconstruire le volume d'information initial. Les différents utilisateurs ont de surcroît l'opportunité d'utiliser n'importe quel sous-ensemble de M projections parmi N. En utilisant de manière idoine la topologie du réseau chaque utilisateur final bénéficiera de M chemins différents adaptés à la position des M versions dégradées de la source. Un exemple de topologie est représentée à la figure 8.1. Trois serveurs comportent trois projections de l'ensemble de la requête. Chaque destinations D_i reçoit un sous-ensemble de quatre projections en provenance des deux serveurs les plus proches alors que juste trois projections sont nécessaires (un sous-ensemble reconstructible et une projection de redondance).
Cet exemple simple révèle parfaitement le rôle de distribution du procédé Mojette lorsque les flux réseaux doivent être gérés en globalité. Quatre avantages majeurs sont identifiables par ce système :

- **Répartition de la charge** en différent chemin (*Load Balancing*). Au lieu d'avoir un énorme flux binaire qui emprunte une seule route, plusieurs flux binaires sont engagés dans la transmission par des serveurs à proximité de l'utilisateur. En outre, si les nœuds d'interconnexion sont aptes à dupliquer les paquets (PIM2[35] *protocol* pour IP) une seule version de projection proviendra du serveur pour l'ensemble des requêtes.
- **Possibilité de dégradation progressive** lorsque une congestion survient. Dans une configuration classique en arbre, quand un nœud subit une congestion les feuilles qui y sont référencées ne reçoivent pas assez d'information. En conséquence, si un codage progressif est entrepris à la source le sous-arbre sera entièrement dégradée de la même façon. Ici, les effets de congestion seront différents selon la localisation de la destination, des serveurs et des points d'interconnexion. L'approche graduable proposée dans le chapitre 5 permet de préserver du réseau l'information prioritaire.
- **Facilité de synchronisation** pour la destination. Habituellement, les destinations doivent mettre en œuvre des techniques de synchronisation entre les différents sous-flux pour reconstruire l'information originale. L'algorithme de transformation Mojette inverse peut débuter dès lors qu'une projection est reçue (rappel du chapitre 3, figure 3.12 page 69). En connaissant les N angles de projections, la destination initialise l'ensemble des bins à zéros. La première projection est exploitée pour re-

[35] Parallel Iterative Methods

construire le maximum d'ixels induisant une mise à jour par valeurs négatives dans les bins utilisés des projections manquantes. L'arrivée d'une deuxième projection se traduit par la somme des valeurs des bins avec les mises à jour. Dans cet algorithme, chaque réception de projection est exploitée au maximum avec anticipation sur la prochaine réception.

– **Sécurité** (intégrité de donnée) du système. L'intégrité est la principale caractéristique de QoS pour des systèmes d'information. Afin d'éviter la destruction de l'information originale, un système distribuée est nécessaire. C'est par ailleurs en termes de répartition que s'est bâti l'Internet à l'origine (figure 1.5 page 18).
La redondance naturelle introduite par la transformation Mojette est facilement utilisable pour protéger les données initiales. Dès qu'en un site donné une perte totale d'information est détectée, les projections concernées sont régénérées à partir d'un autre site par l'application successive de la transformation Mojette inverse et directe. Trop d'attaques doivent être entreprises à différents points du réseau au même moment pour détruire irrémédiablement l'information.

8.2 Application de télémédecine

Le système distribué Mojette présenté dans la section précédente peut être appliqué aussi bien à l'échelle du réseau local (LAN), métropolitain (MAN) qu'à grande échelle (WAN) en fonction de la localisation des différents utilisateurs. Dans l'exemple d'architecture précédente, l'accessibilité (compromis temps-réel et intégrité) était la principale fonctionnalité optimisée. Ici, la sécurité représente en plus un objectif crucial dans l'échange de données médicales qui doivent être disponibles à tout moment. Cependant, l'éclatement des différentes modalités d'acquisitions en imagerie médicale (ultrasons, IRM, scintigraphie...) induisent un stockage d'images déporté sur des sites spécifiques distants. Dans ce contexte, une seule et même gestion peut être pratiquée pour un ensemble donné d'hôpitaux.
Un site distant peut contenir un ensemble de N serveurs contenant chacun une projection de toute la base de données. Il s'agit donc d'associer à un serveur un ensemble d'une projection non reconstructible contrairement à l'architecture de la section précédente. Un (ou plusieurs) serveur contient une projection supplémentaire. L'inaccessibilité d'un serveur (maintenance, panne, remise à jour...), un sous-ensemble de serveurs du site peut satisfaire la requête. Lorsqu'un seul site est utilisé, le mécanisme proposé est déjà plus performant qu'un simple mécanisme RAID (redondance XOR) puisque chaque projection est indépendante des autres et peut être sauvegardé à un deuxième niveau sans compromettre le fonctionnement du système. La situation est d'autant plus avantageuse lorsqu'au moins deux sites sont mis à contribution comme illustré à la figure 8.2.

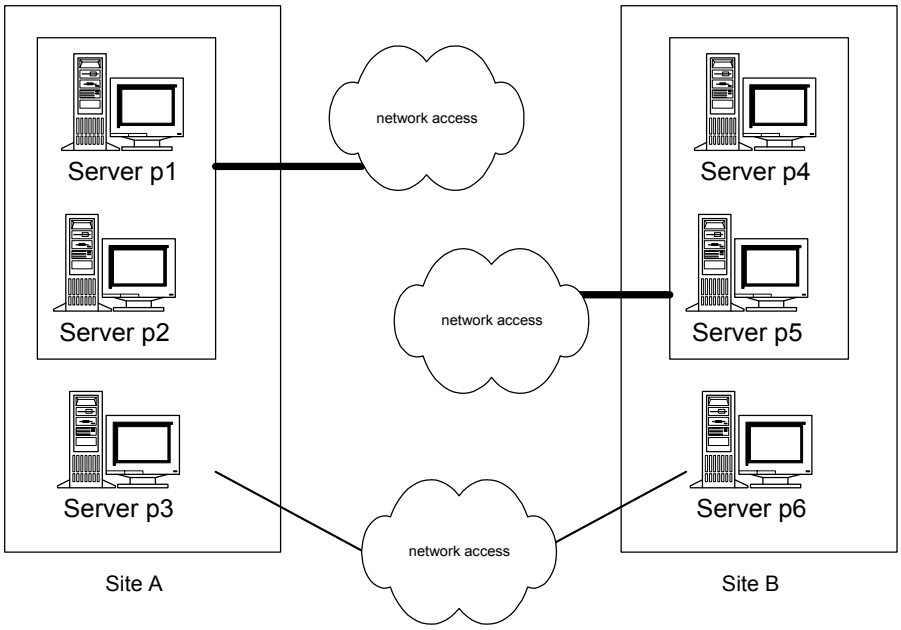


Fig. 8.2: Accès et stockage à une base de données images médicales distribuée sur deux sites.

Dans cet exemple de la figure 8.2, le site A (respectivement le site B) contient trois projections p_1, p_2, p_3 (resp. p_4, p_5, p_6) et deux d'entre elles sont suffisantes pour la reconstruction d'une image ou d'un volume e.g. p_1 et p_2 pour le site A (resp. p_4 et p_5 pour le site B). Maintenant, la projection p_3 (resp. p_6) peut être utilisée pour la redondance locale (pendant le calcul des projections p_1 ou p_4) et pour une reconstruction distante. En empruntant un chemin distinct des autres projections, l'utilisation de cette projection préserve le réseau IP d'une éventuellle congestion. Dans l'objectif de transmettre des informations médicales aux hôpitaux, trois envois séparés doivent être exécutés au même moment à partir des sites A et B.
La mise à jour de la base de donnée s'effectue en isolant plusieurs serveurs du système et ce, en le maintenant disponible. L'agression d'un ou de deux serveurs simultanément ne permet pas la reconstruction même dégradée des données initiales.
Cette architecture distribuée peut être exploitée en commerce électronique pour la mise à jour de comptes utilisateurs à chaque nouvelle transaction.

8.3 Conclusion

Les deux applications exposées dans ce chapitre tiennent compte des aspects d'intégrité et de disponibilité de l'information. La première section décrit un système d'information distribuée sur IP en utilisant la transformation Mojette. La deuxième application, spéci-

fique à la télémédecine, précise le fonctionnement d'une architecture distribuée Mojette dans le cadre d'échange de données médicales. Les systèmes distribués obtenus sont sécurisés, réduisent les congestion sur IP par une répartition naturelle de charge (*load balancing*) et permettent une dégradation progressive de l'information en cas de perte.

Conclusion de la troisième partie

Dans cette dernière partie, nous avons présenté deux types d'applications dans le domaine des transmissions multimédias (chapitre 7) et dans le domaine du stockage de données (chapitre 8). Ces travaux on été réalisés dans deux environnements différents. Le protocole de transfert ionosphérique est issue des recherches menées conjointement par les deux laboratoires de Nantes et de Poitiers donnant lieu à la soutenance d'une thèse et à deux diplôme d'études approfondies.
Les exemples se distinguent entre eux par les données manipulées (images, données médicales) et par les conditions de transmission. Cependant, le contrôle d'intégrité apporté par la description multiple Mojette tenant compte à la fois des propriétés de la source et des caractéristiques du canal permet de s'affranchir de la diversité des paramètres de communication. On peut dégager principalement quatre avantages majeurs de l'ensemble de ces applications :

- **Répartition de la charge** ou *load balancing.* Au lieu d'avoir un énorme flux binaire qui emprunte une seule route, plusieurs flux binaires correspondant à une projection ou à des sous-ensemble de projections sont engagés dans la transmission du signal. L'adéquation entre une projection et un canal permet alors de multiplier les chemins qu'adopte chaque description. Dans le cas de la diffusion, les nœuds d'interconnexion aptes à dupliquer les descriptions réduisent les débits d'injection de paquets au niveau de la source. En stockage, le modèle centralisé organisé autour des *mainframes* (véritable fort Knox de l'information) peut être remplacé par un modèle distribué via la description multiple Mojette. Le nouveau modèle répartit en différents lieux de stockage l'information en réduisant le ratio francs par giga octets. Les propriétés de temps réel et de protection de l'opération Mojette facilite la reprise du contrôle des données éparpillées sur différents sites.
- **Dégradation progressive** en cas d'indisponibilité de l'information (congestion, maintenance d'un serveur de l'architecture distribuée). Dans une configuration classique en arbre, quand un nœud subit une congestion, les feuilles qui y sont référencées perdent une même quantité d'information. La répartition de charge associée à la protection de la graduabilité de la source par description multiple permet de préserver du réseau l'information prioritaire. La réception sera alors fonction de la capacité

du lien avec le réseau de chaque destinataire et de la hiérarchie de la source. Dans un environnement à perte, le système distribue le contenu multimédia de la source à une plus faible résolution.

- **Synchronisation** pour la destination. Classiquement, les destinations doivent mettre en œuvre des techniques de synchronisation entre les différents sous-flux pour reconstruire l'information originale. La taille fixe des projections simplifie le travail à la destination (e.g. protocole de transfert ionosphérique). Le parcours de paquet en paquet reçue permet de localiser la longueur d'un effacement. En outre,connaissant les paramètres de projections (angles, et forme de la mémoire tampon géométrique), la destination est à même d'entreprendre une reconstruction asynchrone. Dans ce cas, chaque arrivée de projection est exploitée au maximum avec anticipation sur la prochaine réception.
- **Sécurité** du système. La redondance introduite par le codage à description multiple Mojette et l'architecture distribuée proposée préserve l'intégrité, caractéristique principale de QoS pour les systèmes d'information, à la fois du réseau en cas de congestion et des attaques malveillantes. Trop d'attaques doivent être menées simultanément en différents points du réseau pour détruire irrémédiablement l'information. La régénération des projections peut être effectuée grâce à d'autres sites préservés ou à d'autres serveurs en fonctionnement (cas de l'application de télémédecine). En cas de congestion, la protection graduable donne lieu à la conservation des niveaux de priorité de la source même si ce canal ne dispose pas de traitements hiérarchisés. Le volume d'information est dégradée progressivement.

La description multiple de l'information sous-entend que le volume de donnée est relativement réduit. La totalité d'une séquence vidéo ne peut dès lors être protégée par ce type de codage. Seule une portion du flux source est sujet à une représentation multiple : le nombre de description est alors relativement faible tout comme leur longueur. Dans ce cas, la description multiple à contraintes temporelles apportent des éléments de réponse. L'usage classique des supports d'information 2D peut être secondé par des mémoires tampon de très grande capacité 3D ou n-D. Les bins générées de cette description sont fortement corrélés car issues de la totalité du volume d'information. Ce faible apport d'information pourrait néanmoins suffire pour achever la reconstruction des supports 2D. Cette solution hybride 2D n-D ouvre les possibilités de protéger par description multiple déterministe des volumes importants de données telles qu'en vidéo ou en imagerie médicale.

Conclusion et perspectives

La description multiple par transformation Mojette se caractérise par des projections géométriques équivalentes d'un support d'information. Le nombre de projections nécessaires à la reconstruction progressive du support permet d'affecter des niveaux de protection aux données sources. Pour des données prioritaires, le support nécessite peu de projections. En revanche, la reconstruction des données d'importance secondaire en requiert un plus grand nombre. Cette approche est utilisée dans le cadre de la transmission et du stockage de données hiérarchisées sur réseau à qualité de service non garantie tel que l'Internet.

Le codage se distingue des méthodes précédentes de description multiple, présentées au chapitre 2, par la simplicité des opérateurs Mojette direct et inverse qui n'utilisent que l'addition et la soustraction sur l'ensemble des entiers. Son principe ainsi que les conditions de reconstruction sont expliqués au chapitre 3. Par ce procédé, il est toujours possible de calculer une projection supplémentaire redondante par rapport à un sous-ensemble reconstructible.

Dans ce contexte, nous avons présenté au chapitre 4 une première méthode de protection uniforme telle qu'un nombre M de projections parmi N soit nécessaire pour reconstruire en totalité le support d'information. L'équivalence des descriptions est déterminée par un choix idoine du support et des angles de projection. La forme hexagonale est retenue pour la taille constante des projections qu'elle implique. Dans son application en transmission de données, cette propriété facilite la synchronisation et équilibre les débits dans le cas d'une répartition de charge (*load balancing*).

Le chapitre 5 étend l'approche au cas de l'information graduable. La source d'information est structurée et délivre un ensemble de sous-flux de priorités distinctes. Un seul support, de forme convexe, est utilisé pour l'ensemble des niveaux de protection. Les sous-flux source sont multiplexés sur le support qui se comporte dès lors comme une mémoire tampon géométrique. La localisation spatiale des zones afférentes aux degrés de protection est détaillée en particulier au niveau des frontières entre ces zones. Une fois délimitée, ces zones de protection délivrent de manière déterministe la même quantité d'information pour un nombre donné de projections. Les projections sont effectuées sur le support entier. Le choix d'un seul support pour protéger une source graduable permet de réduire les correspondances explicites entre un élément de projection (bin) et une donnée (ixel) et donc

le coût inhérent à l'algorithme Mojette inverse. Dans ce cas, la solution optimale MDS est approchée au profit d'une complexité réduite. On montre de plus que le fait d'augmenter la capacité du support d'information (volume source important ou représentation binaire) optimise de même le décodage en diminuant la proportion des bins supplémentaires.
Chaque description possède une fraction de flux qui est décodable progressivement dès que les conditions de reconstruction différentes pour chaque sous flux sont remplies. Si la reconstruction est hiérarchisée en relation avec la structure de la source, les projections entre elles ne présentent aucune distinction. L'équivalence des pouvoirs de reconstruction des projections, la reconstruction progressive et la taille constante des projections font cette deuxième méthode de protection inégale un codage à description multiple équilibrée. Les niveaux de protection sont attribués à chaque sous-flux en fonction des propriétés de la source (incréments de qualité de chaque flux) et des caractéristiques du canal (probabilité de perte de paquets). Le schéma de codage conjoint source-canal obtenu au chapitre 6 permet de résoudre le problème formulé en introduction, celui de la transmission d'une source hiérarchisée sur un canal de type "best effort" où les pertes se produisent aléatoirement. On effectue, via cette méthode, la transmission de la structure hiérarchique de la source de manière transparente pour le canal. La perte des descriptions entame progressivement la qualité de la reconstruction. Une version dégradée de la source est toujours reçue, même pour un taux de perte élevé.

Cette approche de protection inégale est comparée au cas égal. Le gain en qualité est manifeste pour la solution optimisée de redondance multi-niveaux. La stratégie "tout ou rien" de la protection égale n'est pas adaptée aux sources actuelles ni aux environnements à pertes de paquets. En regard de la structure hiérarchisée, un comportement progressif dans l'introduction de la redondance est plus adéquat. Son optimisation constitue un codage conjoint source-canal où le canal est défini en prenant en compte, en plus des caractéristiques de la transmission, les organes d'émission et de réception. Englobant les aspects protocolaires inhérents au transport de données, notre codage apporte la robustesse nécessaire au codeur source pour la transmission de ses éléments sur un réseau à qualité de service non garantie.
Par les ixels que l'opérateur Mojette manipule, une source réelle quelconque d'information quantifiée peut être protégée (JPEG progressif). Le chapitre 7 fait état de la transmission de données issue d'un codeur SPIHT. D'autres sources d'images (JPEG-2000) et de vidéo (MPEG) sont supportées dès lors qu'une structure hiérarchisée est à disposition de notre codeur. Le canal emprunté fait référence principalement au protocole de l'Internet avec en couches supérieures, un protocole de contrôle tel que RTCP ou un mécanisme d'adaptatif comme TCP-friendly qui permettent d'obtenir, en voie de retour, les méta données utiles à l'allocation de la redondance (probabilité de perte, délais). De même qu'en transmission, l'application de la description multiple par transformation Mojette au stockage de données dans une architecture distribuée présente des avantages significatifs (chapitre 8) en

termes de sécurité (intégrité de donnée et régénération de l'information par projection), de répartition des charges, de disponibilité (possibilité d'obtenir une version dégradée dans un court délai) et de reprise de contrôle sur les données éparpillées.

En perspective de ce travail, une version probabiliste du système est envisagée pour des grandes quantités de données telles que la vidéo ou encore les images médicales. La description ne peut pas dans ce cas s'effectuer sur la totalité du volume d'information et respecter les contraintes de temps réel de ce type d'application (temps de remplissage des mémoires tampon géométriques). On effectue alors la description multiple morceau par morceau. Les paquets générés ne sont pas équivalents par les portions différentes du flux qu'ils décrivent. Le système est dans ce cas probabiliste dans la mesure où la réception d'un nombre de paquets sur l'ensemble envoyé ne donne pas lieu à une valeur certaine de distorsion. Le travail se situe donc au niveau du découpage séquentiel sur lequel il faudra effectuer la description multiple. Par ailleurs, à ce découpage séquentiel peut être ajoutée une description multiple à contrainte temporelle (introduite au chapitre 7). Chaque séquence est à la fois codée par la description multiple d'un espace 2D vers un ensemble de signaux 1D et par une mémoire tampon géométrique de dimension N (de grande capacité) offrant des descriptions supplémentaires qui corrèlent l'ensemble des données.

L'approche multidimensionnelle permet de résoudre le problème du nombre de descriptions. Dans le cas présenté, la redondance introduite est fonction du nombre de projections (fonction plus exactement de la somme du paramètre d'angle p). Une projection supplémentaire redondante pour un sous-ensemble reconstructible peut toujours être calculée mais augmente le paramètre d'angle p et donc la redondance. Le passage direct d'un espace 3D vers des signaux 1D pourrait compenser ce coût inhérent à la dispersion de l'information. Il convient de respecter dans le dimensionnement de ces mémoires tampon géométriques 3D, voire ND, le compromis contrôle d'intégrité (taux de redondance) et contrainte de temps réel (taille de paquets). Dans le contexte de la réduction du coût de la dispersion d'information, la compression des descriptions est aussi sujette à une étude plus approfondie.

Au niveau du canal, le modèle de perte reste générique. La loi exponentielle ne représente que de façon approchée les pertes qui peuvent survenir sur un canal tel que l'Internet. L'optimisation sur des méta données réelles caractérisant le canal (taux de perte de paquets, délais) permettrait une meilleure adaptation de notre système de protection. Par les informations de la source et du canal, la mise en forme dynamique des mémoires tampon géométriques doit pouvoir offrir une réaction adéquate de la source à l'environnement de transmission.

Une nouvelle optimisation de ce système est envisagée en considérant davantage les propriétés de la source et les caractéristiques du canal. Pour la source, de nouveaux critères de qualité seraient à considérer, comme l'évaluation subjective par un critère objectif de la qualité. Les niveaux de protection seraient de ce fait issus de la perception psychovisuelle

du signal d'image et apporteraient une contribution dans la préservation de la sémantique même de la source.

Annexe A

Mise en œuvre de la Mojette XOR

La mise en œuvre de la Mojette est considérablement simplifiée dans le cas où :

les éléments d'informations (ixels) sont quantifiés sur un bit. L'opération d'addition est alors l'opérateur OU EXCLUSIF (XOR) ou encore l'addition dans GF(2).

Le paramètre q de l'angle Mojette est égale à 1. Un bin correspond alors à la somme d'un ixel de chaque ligne de la mémoire tampon géométrique.

La figure A.1 ci-contre distingue trois cas dans le calcul des projections (0,1), (1,1) et (-1,1) selon que p est nul, positif ou négatif.

Cas où $p = 0$
À une colonne coïncide une somme (un bin). La projection (0,1) est un simple contrôle de parité pour chaque colonne du support d'information (*check sum*).

Cas où $p > 0$
La figure illustre ce cas par le calcul de la projection (1,1). Dans ce cas, la première ligne du tableau en partant du haut subit un décalage à gauche de p bits. Un XOR entre la première ligne et de la deuxième ligne est effectué. On décale à nouveau les valeurs de cette nouvelle ligne de p bits et on effectue le XOR de cette deuxième ligne décalée avec la troisième ligne du support. On reitère l'opération jusqu'à obtenir la projection (au bout de la quatrième ligne dans l'exemple).

Cas où $p < 0$
La méthode est similaire au cas précédent avec la première ligne du support en commençant par le bas.

Dans les deux cas où p est non nul, il convient de vérifier dans l'allocation que les décalages successifs n'engendre pas de perte de données.

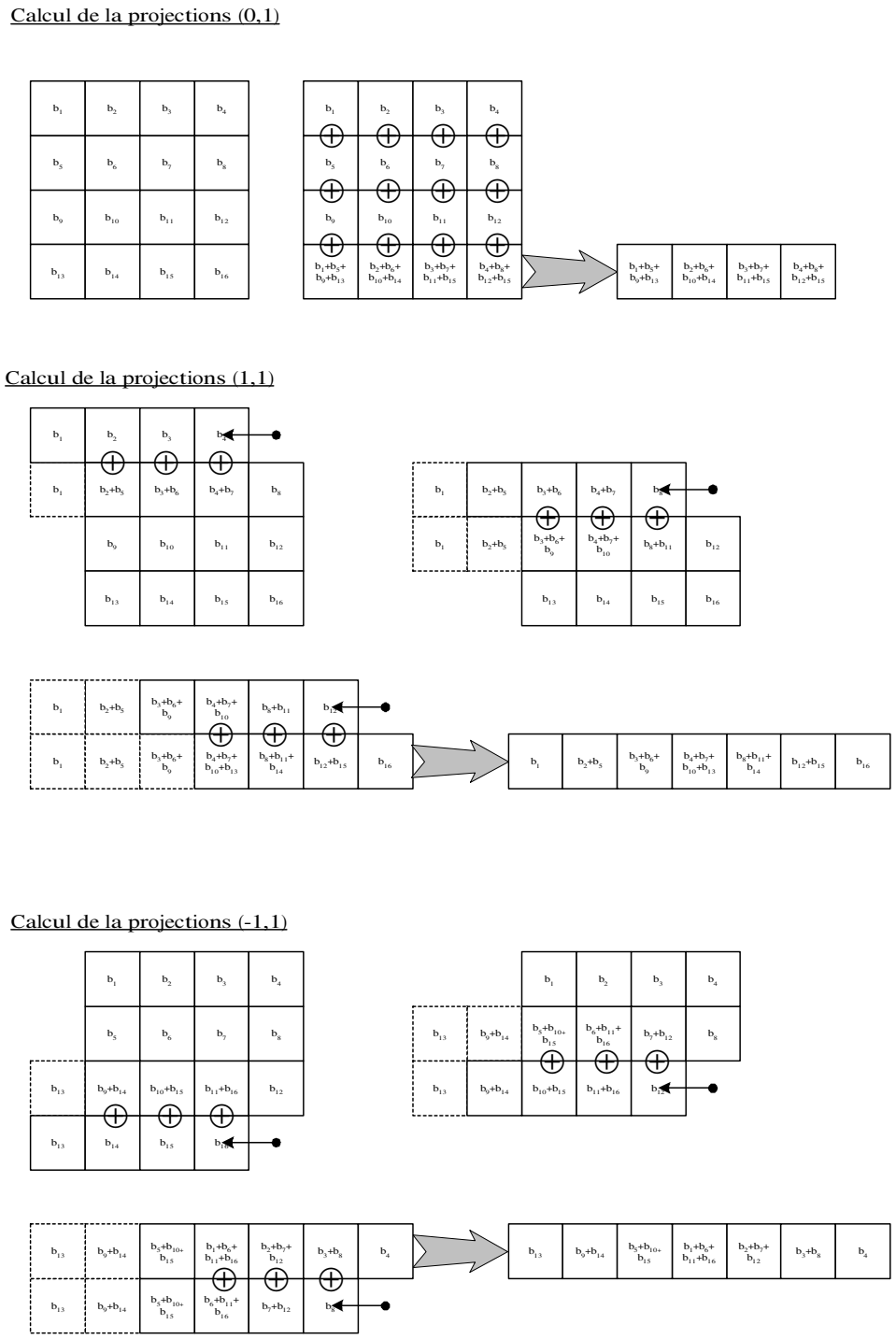


Fig. A.1: Mise en œuvre de la transformation Mojette XOR au travers du calcul des projections de type (0,1), (1,1) et (-1,1) sur un support binaire d'information de dimension 4×4.

Annexe B

Algorithme de la Mojette inverse

On indique dans cette annexe l'algorithme de Mojette inverse en considérant les trois phases d'initialisation décrites au § 3.2.5 page 59. Pour atteindre une complexité en $\mathcal{O}(IN)$ où I est le nombre d'éléments d'information et N le nombre de projections, il convient d'observer que plusieurs algorithmes ont été mis en œuvre [Guédon 01b].

Algo_MojInv2

Début
Initialisation()
Pour compteur de 1 à PxQ faire
(i,m) $\longleftarrow$ *Phase_1* *//cherche correspondance univoque*
(lin, col) $\longleftarrow$ *Phase_2(i,m)* *//cherche la position du pixel*
image(lin, col) $\longleftarrow$ *proj(i,m)* *//rétroprojection*
Phase_3 *//mise à jour des projections*
FinPour
Fin

avec

Initialisation()

Début
Pour compteur de 1 à N = PxQ faire *//boucle sur le nombre pixels*
lin $\longleftarrow$ *compteur/P ; col* $\longleftarrow$ *compteur - (lin-1).P ; image(lin, col)* $\longleftarrow$ *-1*
Pour i de 1 à I *//boucle sur le nombre de projections*
bin(compteur, i) $\longleftarrow$ $p_i.lin - q_i.col$
FinPour
FinPour *//image initialisée*
//∘∘
Pour i de 1 à I faire
$p \longleftarrow p_i$; $q \longleftarrow q_i$

NbBin(i) $\longleftarrow$ *(P-1).*p_i*+(Q-1).*q_i*+1*
Pour compteur de 1 à NbBins(i) faire
projelem(compteur, i) $\longleftarrow$ *0* *//nombre de pixels qui contribuent à ce bin*
FinPour
FinPour *//projelem est partiellement initialisée*
//°°
Pour i de 1 à I faire
p $\longleftarrow$ p_i *; q* $\longleftarrow$ q_i *; image(lin, col)* $\longleftarrow$ *-1*
Pour compteur de 1 à N=PxQ faire
lin $\longleftarrow$ *compteur/P ; col* $\longleftarrow$ *compteur-(lin-1).P ; compteur=(lin-1).P+col*
m $\longleftarrow$ $p_i.lin - q_i.col$
bin(compteur, i) $\longleftarrow$ *m* *//position du bin pour une projection i et image(lin,col)*
projelem(m,i) +$\longleftarrow$ *1* *//nombre de pixels qui contribuent à ce bin*
projposition(m,i) +$\longleftarrow$ *compteur* *//somme des indices uniques qui contribuent à ce bin*
FinPour
FinPour *//projelem, projposition et bin sont initialisés*
//°°
NbCorresUniv $\longleftarrow$ *0*
Pour compteur de 1 à N=PxQ faire *//boucle sur le nombre de pixels*
lin $\longleftarrow$ *compteur/P ; col* $\longleftarrow$ *compteur-(lin-1).P ; i* $\longleftarrow$ *1 ; pasfini* $\longleftarrow$ *vrai*
Tant que (i≤I et pasfini) faire *//boucle sur projections avec arrêt sur le premier bin trouvé si il existe*
Si (projelem(m=bin(compteur,i))=1) alors
NbCorresUniv +$\longleftarrow$*1* *//remplir le tableau de correspondances univoques*
corresUniv(NbCorresUniv-1)[pixel, projection, bin] $\longleftarrow$ *(compteur, i, m)*
pasfini $\longleftarrow$ *faux*
Sinon
i +$\longleftarrow$*1*
FinSi
FinPour *//corresUniv initialisé*
//°°
Fin

Phase_1

//cherche correspondance univoque : on dépile le dernier
(compteur, i, m) $\longleftarrow$ *one2one(NbCorresUniv-1)[pixel, projection, bin]*
NbCorresUniv -$\longleftarrow$*1*
//Ici soit nous avons trouvé (compteur, i, m) soit la transformation n'est pas inversible

Fin

Phase_2

//reconstruction du pixel qui est défini par (compteur, i, m)

image(compteur $\longleftarrow$ *proj(m,i)*

Fin

Phase_3

//mise à jour de toute les projections

Pour i de 1 à I faire

$m \longleftarrow p_i.lin - q_i.col$ *//calculer bin m correspondant au pixel reconstruit*

proj(i,m) -$\longleftarrow$ *image(lin, col)* *//décrément le bin*

projelem(i,m) -$\longleftarrow$ *1* *//décrément le #élements*

projposition(i,m) -$\longleftarrow$ *compteur* *//décrément du compteur de ce bin*

if (projelem(i,m)=1) then *//ajout dans tableau des correspondances univoques*

NbCorresUniv +$\longleftarrow$ *1*

corresUniv(NbCorresUniv-1)[pixel, projection, bin] $\longleftarrow$ *projposition(i,m)*

FinSi

FinPour

Fin

Bibliographie

[Alatan 00] A.A. Alatan, M. Zhao & A. N. Akansu. *"Unequal Error Protection of SPIHT Encoded Image Bit Streams"*. IEEE Journ. on Sel. Areas in Commun., vol. 18, no. 6, pages 814–818, Juin 2000.

[Albanese 96] A. Albanese, J. Blömer, J. Edmonds, M. Luby & M. Sudan. *"Priority Encoding Transmission"*. IEEE Trans. on Inform. Th., vol. 42, no. 6, pages 1737–1744, Novembre 1996.

[Alon 96] N. Alon & M. Luby. *"A Linear Time Erasure-Resilient Code with Nearly Optimal Recovery"*. IEEE Trans. on Inform. Th., vol. 42, no. 6, pages 1732–1736, Novembre 1996.

[Arean 00] R. Arean, J. Kovačević & V.K. Goyal. *"Multiple Description Perceptual Audio Coding with Correlated Transforms"*. IEEE Trans. on Speech and Audio Proc., vol. 8, no. 2, pages 140–145, Mars 2000.

[Autrusseau 01] F. Autrusseau & JP. Guédon. *"Tatouage Mojette D'images Médicales Dans un Environnement Distribué Avec Tiers de Confiance"*. Traitement du Signal, 2001. soumission mars 2001.

[Baran 64] P. Baran. *"On Distributed Communications : IV. Priority, Precedence, and Overload"*. Rapport Technique RAND corporation, Santa Monica (CA), no. RM-3638-PR, Août 1964.

[Batllo 97] J.-C. Batllo & V. A. Vaishampayan. *"Asymptotic Performance of Multiple Description Transform Codes"*. IEEE Trans. on Inform. Th., vol. 43, no. 2, pages 703–707, 1997.

[Bennani 01] F. Bennani, A. Boutignon & N. Simoni. *"IP QoS Issues : Efficient Cooperation of Diffserv, MPLS, and Management"*. SPIE ITCom'01, vol. 4524, page 11 pp., Août 2001.

[Blömer 95] J. Blömer, M. Kalfane, R. Karp, M. Karpinski, M. Luby & D. Zuckerman. *An Xor-Based Erasure-Resilient Coding Scheme*. Rapport technique TR-95-048, International Computer Science Institute, 1995.

[Braden 97] R. Braden, L. Zhang, S. Berson AndI S. Herzog & S. Jamin. *"Ressource Reservation Protocol RSVP"*. Network Working Group RFC 2205, Septembre 1997.

[Byers 98] J. W. Byers, M. Luby, M. Mitzenmacher & A. Rege. *A Digital Fountain Approach to Reliable Distribution of Bulk Data*. Rapport technique TR-98-013, International Computer Science Institut, Berkeley, California, 1998.

[Calderbank 98] A. R. Calderbank. *"The Art of Signaling : Fifty Years of Coding Theory"*. IEEE Trans. on Inform. Th., vol. 44, no. 6, pages 2561–2595, 1998.

[Camredon 01] B. Camredon & N. Herreyre. *"Architecture de Micro-Paiement NetCard"*. projet SILR2, Juin 2001.

[Chou 99] P. Chou, S. Mehrotra & A. Wang. *"Multiple Description Decoding of Overcomplete Expansions Using Projections Onto Convex Sets"*. Proc. IEEE Data Compr. Conf., Snowbird, (UT), pages 72–81, 1999.

[Clark 90] D. D. Clark & D. L. Tennenhouse. *"Architectural Considerations for a New Generation of Protocols"*. dans Proc. ACM SIGCOMM, 1990.

[Cosman 00] P.C. Cosman, J.K. Rogers, P.G. Sherwood & K. Zeger. *"Combined Forward Error Control and Packetized Zerotree Wavelet Encoding for Transmission of Images Over Varying Channels"*. IEEE Transactions on Image Processing, vol. 9, no. 6, pages 982–993, june 2000.

[Davis 96] G. Davis & J. Danskin. *"Joint Source and Channel Coding for Image Transmission over Lossy Packet Networks"*. Proc. SPIE Appl. Digital Image Processing XIX, pages 376–387, Août 1996.

[Deering 91] Stephen E. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, Décembre 1991.

[Deering 93] Steve Deering. *"Internet Multicast Routing : State of the Art and Open Research Issues"*. Multimedia Integrated Conferencing for Europe (MICE), Séminaire au Swedish Institute of Computer Science, Stockholm, Octobre 1993.

[Duhamel 97] P. Duhamel & O. Rioul. *"Codage Conjoint Source/Canal : Enjeux et Approches"*. GRESTI, pages 699–704, Septembre 1997.

[El-Gamal 82] A. A. El-Gamal & T. Cover. *"Achievable Rates for Multiple Descriptions"*. IEEE Trans. on Inform. Th., vol. 28, no. 6, pages 851–857, 1982.

[Equitz 91] W. H. R. Equitz & T. M. Cover. *"Successive Refinement of Information"*. IEEE Trans. on Inform. Th., vol. 37, no. 2, pages 269–275, mars 1991.

[Escrig 97] B. Escrig. *Estimation Des Paramètres D'un Modèle de Canal Ionosphérique*. thèse de Doctorat, université de Poitiers, Décembre 1997.

[Garrett 93] M. W. Garrett & M. Vetterli. *"Joint Source/Channel Coding of Statistically Multiplexed Real-Time Services over Packet Networks"*. IEEE/ACM TRans. on Networking, vol. 1, no. 1, pages 71–80, 1993.

[Goyal 98a] V. K. Goyal & J. Kovačević. *"Optimal Multiple Description Transform of Gaussian Vectors"*. Proc. IEEE Data Compr. Conf., Snowbird, (UT), pages 388–397, Snowbird, UT, Mars 1998.

[Goyal 98b] V. K. Goyal, J. Kovačević, R. Arean & M. Vetterli. *"Multiple Description Transform of Coding Images"*. Proc. IEEE Int. Conf. Image Processing, Chicago, IL., 1998.

[Goyal 01a] V. K. Goyal & J. Kovačević. *"Generalized Multiple Description Coding with Correlating Transforms"*. IEEE Trans. on Inform. Th., 2001. à paraître.

[Goyal 01b] V.K. Goyal. *"Muliple Description Coding : Compression Meets the Network."*. IEEE Signal processing magazine, 2001. en préparation.

[Group 98] Ind. JPEG Group. *Version 6b de Cjpeg.* ftp ://ftp.uu.net/graphics/jpeg, 1998.

[Guédon 95a] JP. Guédon & D. Barba. *"Décomposition Psychovisuelle et Transmission Rapide D'images"*. CORESA, page 6 pages, Janvier 1995.

[Guédon 95b] JP. Guédon, D. Barba & N. Burger. *"Psychovisual Image Coding Via an Exact Discrete Radon Transform"*. Visual Communication and Image Representation, 1995.

[Guédon 01a] J. P. Guédon, B. Parrein & N. Normand. *"Internet Distributed Image Information System"*. Integrated Computer-Aided Engineering, vol. 8, pages 205–214, 2001.

[Guédon 01b] JP. Guédon. *Histoire de la Transformation Mojette (la Forme et la Projection).* préparation de l'Habilitation à Diriger des Recherches, 2001. université de Nantes.

[Hénocq 00] X. Hénocq, F. Le Léannec & C. Guillemot. *"Joint Source and Channel Rate Control in Multicast Layered Video Transmission"*. SPIE Visual Communications and Image Processing, vol. 4067, pages 296–307, 2000.

[ITU 00] ITU. *"Méthodologie D'évaluation Subjective de la Qualité Des Images de Télévision"*. ITU-R Recommendation BT. 500-10, 2000.

[ITU 01] Secteur de la Normalisation Des Télécommunications ITU. *"Results of an Audiovisual Desktop Video Teleconferencing Subjective Experiment"*. Commission d'étude 12, 2001.

[Jacobson 88] Van Jacobson. *"Congestion Avoidance and Control"*. Proc. ACM of SIGCOMM'88, Stanford (CA), pages 316–329, Août 1988.

[Jiang 99] W. Jiang & A. Ortega. *"Multiple Description Coding Via Polyphase Transform and Selective Quantization"*. SPIE Visual Communications and Image Processing, San José, CA, vol. 3653, pages 998–1008, Janvier 1999.

[JPEG2000] JPEG2000. *"Coding of Still Picture. JPEG2000 Verification Model 5.3 (Technical Description)"*. ISO/IEC JTC 1/SC 29/WG 1 N1450, Novembre, 1999.

[Katz 79] M. Katz. *"Questions of Uniqueness and Resolution in Reconstruction Form Projections"*. Lectures Notes in Biomathematics, vol. 26, 1979.

[LeCallet 01] P. LeCallet & D. Barba. *"Image Quality Assesment : From Site Errors to a Global Appreciation of Quality"*. Picture Coding Symposium, Séoul, pages 105–108, 2001.

[Leicher 94] C. Leicher. *"Hierarchical Encoding of Mpeg Sequences Using Priority Encoding Transmission (Pet)"*. Rapport technique TR-94-058, ICSI, Berkeley, CA, Novembre 1994.

[LeLéannec 01] F. LeLéannec. *Codage Vidéo Robuste et Hiérarchique Pour la Transmission sur Réseaux Hétérogènes*. PhD thesis, Université Rennes 1, école doctorale MATISSE, Février 2001.

[Liang 99] J. Liang & R. Talluri. *"Tools for Robust Image and Video Coding in JPEG-2000 and MPEG-4 Standards"*. Proc. SPIE Visual Commun. Image Processing", vol. 3653, pages 40–51, 1999.

[Luby] M. Luby. *"Cauchy-Based Reed Solomon Codes"*. disponible à http ://www-nrg.ee.lbl.org/ luby.

[Luby 97] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman & V Stemann. *"Pratical Loss-Resilient Codes"*. Proceedings of the 29th ACM Symposium on Theory of computing, 1997.

[Madhavi 97] J. Madhavi & S. Floyd. *"TCP-Friendly Unicast Rate-Based Flow Control"*. Technical note sent to the end2end-interest mailing list, Janvier 1997.

[Mallat 89] S. G. Mallat. *"A Theory for Multiresolution Signal Decomposition : The Wavelet Representation"*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pages 674–693, Juillet 1989.

[Mathis 97] M. Mathis, J. Semke, J. Madhavi & T. Ott. *"The Macroscopic Behavior of the TCP Congestive Avoidance Algorithm"*. CCR, vol. 27, no. 3, Juillet 1997.

[McCanne 96] S. R. McCanne. *Scalable Compression and Transmission of Internet Multicast Video.* PhD thesis, Université de Californie Berkeley, Décembre 1996.

[Miguel 99] A. C. Miguel, E. A. Riskin & A. E. Mohr. *"SPIHT for Generalized Multiple Description Coding".* Proc. IEEE Conf. on Image Proc., 1999.

[Miller 78] S.E. Miller. *"New Transmission Configuration"*, 1978. Cahier Laboratoire.

[Moccagatta 00] I. Moccagatta, S. Soudagar, J. Liang & H. Chen. *"Error-Resilient Coding in JPEG-2000 and MEPG-4".* IEEE Journ. on Sel. Areas in Commun., vol. 18, no. 6, pages 899–914, Juin 2000.

[Mohr 00] A. Mohr, E. A. Riskin & R. E. Ladner. *Unequal Loss Protection : Graceful Degradation of Image Quality over Packet Erasure Channels Through Forward Error Correction.* Journal on Sel. Areas in Com., vol. 18, no. 6, pages 819–828, Juin 2000.

[Moreau 95] N. Moreau. *Techniques de Compression Des Signaux.* Chapitre 5 : codage par transformée, codage en sous-bandes, pages 129–150, 1995. Paris, Masson.

[Ng 97a] F. Ng & J. Kovačević. *Nonredundant Image Representations.* Proc. IEEE Conf. on Image Proc., vol. 2, pages 434–437, Santa Barbara, CA, Octobre 1997.

[Ng 97b] F. Ng & J. Kovačević. *Raw Image Data Transmission over Lossy Packet Networks.* soumis à IEEE Trans. on Image Processing, Août 1997.

[NOC] NOC. *Internet Weather Report.* disponible à http ://www.noc.ucla.edu/networking/weather.html.

[Normand 97] N. Normand. *Représentation d'Images et Distances Discrètes Basées sur Les Elements Structurants À Deux Pixels.* thèse de Doctorat, Sciences pour l'Ingenieur de Nantes, jan. 1997.

[Orchard 97] M. T. Orchard, Y. Wang, V. Vaishampayan & A. R. Reibman. *"Redundancy Rate-Distorsion Analysis of Multiple Description Coding Using Pairwise Correlating Transforms".* Proc. IEEE Conf. on Image Proc., vol. 1, no. 3, pages 608–611, Santa Barbara (CA), Octobre 1997.

[Ortega 98] A. Ortega & K. Ramchandran. *"Rate-Distorsion Methods for Image and Video Compression".* IEEE Signal Proc. Mag., pages 23–50, Novembre 1998.

[Ozarow 80] L. Ozarow. *"On a Source-Coding Problem with Two Channels and Three Receivers".* Bell Syst. Journal, vol. 59, no. 10, pages 1909–1921, Décembre 1980.

[Pakosz 01] G. Pakosz & C. Meyer. *"Protocole de Transmission Multimédia À Travers L'ionosphère"*. projet SILR2, 2001.

[Parrein 01a] B. Parrein, N. Normand & J. P. Guédon. *"Multiple Description Using Exact Discrete Radon Transform"*. Proc. IEEE Data Compr. Conf., Snowbird, (UT), page 508, mars 2001.

[Parrein 01b] B. Parrein, P. Verbert, N. Normand & JP. Guédon. *"Scalable Multiple Descriptions on Packets Networks Via the N-Dimensional Mojette Transform"*. SPIE ITCom'01, vol. 4524, pages 243–252, 2001.

[Paxson 97] V. Paxson & S. Floyd. *"Why We Don't Know How to Simulate the Internet"*. in the Proceeding of the 1997 Winter Simulation Conference, Atlanta, GA, 1997.

[Philippé 98] O. Philippé. *Représentation D'images Pour Le Codage Conjoint Source-Canal sur Réseaux À Qualité de Service*. PhD thesis, Université de Nantes, IRESTE, 1998.

[Pouliquen 00] C. Pouliquen. *Méthode de Transmission Confidentielle et Sûre sur Réseaux de Paquets*. Mémoire CNAM, Mars 2000.

[Radon 17] J. Radon. *"Über Die Bestimmung Von Functionen Durch Ihre Integralwerte Langs Gewisser Mannigfaltigkeiten"*. Berichte sachsische academie der wissenchaften, Leipzig, Math.-Phys. K1, vol. 69, pages 262–267, 1917.

[Reudink 80] D.O. Reudink. *The Channel Splitting Problem with Interpolative Coders*. Rapport technique TM80-134-1, Bell Laboratories, Octobre 1980.

[Rizzo 97] L. M. Rizzo. *"A Reliable Multicast Data Distribution Protocol Based on Software FEC Techniques"*. Computer Communication Review, Avril 1997.

[Said 96a] A. Said & W. A. Pearlman. *"A New, Fast, and Efficient Image Codec Based on Set Partitionning in Hierarchical Trees"*. IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, no. 3, pages 243–249, Juin 1996.

[Said 96b] A. Said & W. A. Pearlman. *Codeur SPIHT*, 1996. disponible à http ://www.cipr.rpi.edu/research/SPIHT/.

[Savage 99] S. Savage. *"Sting : A TCP-Based Network Measurement Tool"*. Proc. 1999 USENIX Symp. Internet Technol. Syst., pages 71–79, Octobre 1999.

[Schooler 97] E. Schooler & J. Gemmell. *Using Multicast FEC to Solve the Midnight Madness Problem*. Rapport technique MS-TR-97-25, Microsoft Research, Septembre 1997.

[Schulzrinne 96] H. Schulzrinne, S. Casner, R. Frederick & V. Jacobson. *"RTP : A Transport Protocol for Real-Time Applications"*. IETF RFC 1889, Janvier 1996.

[Servetto 99a] S. Servetto. *Compression and Reliable Transmission of Digital Image and Video Signals*. PhD thesis, University of Illinois at Urbana-Champaign, 1999.

[Servetto 99b] S.D. Servetto & K. Nahrstedt. *"Video Streaming over the Public Internet : Multiple Description Codes and Adaptative Transport Protocols"*. Proc. IEEE Conf. on Image Proc., Kobé, Japon, Octobre 1999.

[Servetto 00] S.D. Servetto, K. Ramchandran, V. A. Vaishampayan & K. Nahrstedt. *"Multiple Description Wavelet Based Image Coding"*. IEEE Trans. on Image Processing, vol. 9, no. 5, pages 813–825, Mai 2000.

[Srivinasan 98] M. Srivinasan & R. Chellappa. *"Multiple Description Subband Coding"*. Proc. IEEE Conf. on Image Proc., Chicago, IL 1998.

[Tan 99] W. Tan & A. Zakhor. *"Internet Video Using Error Resilient Scalable Compression and Cooperative Transport Protocol"*. Proc. IEEE Conf. on Image Proc., Octobre 1999.

[Vaishampayan 93] V. A. Vaishampayan. *Design of Multiple Description Scalar Quantizers*. IEEE Trans. Information Theory, vol. 39, no. 3, pages 821–834, mai 1993.

[Verbert 01] P. Verbert & JP. Guédon. *"Transformation Mojette En Dimension N"*. cd-rom GRETSI'01, Toulouse, no. 45, Septembre 2001.

[Wang 97] Y. Wang, M.T Orchard & A. Reibman. *"Multiple Description Image Coding for Noisy Channel by Pairing Transform Coefficients"*. Proc. 1st Workshop IEEE on Multimedia Signal Proc, pages 419–424, Princeton, NJ, Juin 1997.

[Wang 98] Y. Wang, M. Orchard & A. Reibman. *Optimal Pairing Correlating Transforms for Multiple Descriptions Coding*. Proc. IEEE Conf. on Image Proc., pages 679–683, Chicago, IL, Octobre 1998.

[Watson 93] A. B. Watson. *"DCTune : A Technique for Visual Optimization of DCT Quantization Matrices for Individual Images"*. SID digest of technical papers XXVI, pages 946–949, 1993.

[Weaver 49] W. Weaver & C. E. Shannon. *"Théorie Mathématique de la Communication"*. Les classiques des sciences humaines, 1949. réédition de 1975.

[Wells 96] David Wells. Le dictionnaire penguin des curiosités géométriques. Eyrolles, 1996.

[Wen 98] J. Wen & J. Villasenor. *"Reversible Variable Length Codes for Efficient and Robust Image and Video Coding"*. Proc. IEEE Data Compr. Conf., Snowbird, (UT), pages 471–480, Snowbird (UT), Mars 1998.

[Williams 77] F.J. Mac Williams & N.J.A. Sloane. The theory of error correcting codes. North-Holland Amsterdam-New York-Oxford, 1977.

[Wolf 80] J.K. Wolf, A.D. Wyner & J. Ziv. *Source Coding for Multiple Descriptions*. Bell Syst. Journal, vol. 59, no. 8, pages 1417–1426, octobre 1980.

[Wolf 99] T.Y. Berger Wolf & E.M. Reingold. *"Optimal Index Assignement for Multichannel Communication"*. Proc. SIAM Symp. Discrete Algorithms, Baltimore, MD 1999. Version disponible sur http ://emr.cs.uiuc.edu/ reingold/algorithms.html.

Production scientifique

PUBLICATIONS EN REVUES :

1. **Internet Distributed Image Information System**
   J.P. Guédon, B. Parrein, N. Normand, Integrated Computer-Aided Engineering. vol. 8, n°3, 2001, pp. 205-214.
2. **Post-synchronized scintigraphic data to estimate antral motility**
   C. Le Rest, B. Parrein, V. Morin, B. Bridji, Y. Bizais, O. Couturier, Nuclear Medicine Communications, à paraître.

PUBLICATIONS EN CONGRES INTERNATIONAUX AVEC COMITE DE LECTURE ET ACTES :

1. **Scalable Multiple Descriptions on Packet Networks via the n-dimensional Mojette transform**
   B. Parrein, P. Verbert, N. Normand, J.P. Guédon, SPIE ITCom 2001, pp. 243-252, août 2001, Denver.
2. **Load-balancing and scalable multimedia distribution using the Mojette transform**
   J.P. Guédon, N. Normand, P. Verbert, B. Parrein, F. Autrusseau, SPIE ITCom 2001, pp. 226-234, août 2001, Denver.
3. **Multiple Description Coding using Exact Discrete Radon Transform**
   B. Parrein, N. Normand, JP. Guédon, Data Compression Conference, p.508, mars 2001, Snowbird(UT).
4. **Distributed image transmission and storage on Internet system**
   J.P. Guédon, N. Normand, B. Parrein, C. Pouliquen, ACIDCA 2000, pp. 164-169, vol. VPR, mars 2000, Tunis.

AUTRES PUBLICATIONS :

Conférences nationales avec comité de lecture

1. **Description multiple par transformation de Radon discrète exacte**
   B. Parrein, N. Normand, D. Barba, GRETSI 2001, 4 p., septembre 2001, Toulouse.
2. **Système de transmission par codage prioritaire**
   B. Parrein, N. Normand, J.P. Guédon, CORESA 2000, 8 p., octobre 2000, Poitiers(France).
3. **Description multiple et codage source-canal sur un canal à effacement avec la transformation Mojette**
   N. Normand, B. Parrein, GRETSI 99, pp. 869-872, vol. 3, septembre 1999, Vannes(France).
4. **Multiplexage de flux multimédias sur IP par description multiple Mojette**
   B. Parrein, N. Normand, CORESA 99, 8 p., juin 1999, Sophia-Antipolis(France).

Divers

1. **Transformation Mojette : un outil pour la sécurité**
   B. Parrein, P. Verbert, J.P. Guédon, N. Normand, Journée thématique ISIS : Signal et Image au service de la Sécurité dans la Société de l'Information, octobre 2000, CNRS Centre de Paris-Meudon
2. **Protection de flux applicatifs sur un canal à effacement**
   B. Parrein, Forum jeunes chercheurs ISIS, mai 2000, Paris