

Titre: Compression asymétrique de séquences stéréoscopiques par compensation de mouvement/disparité et modélisation fractale

Auteur: Christine Frobert

Date: 1996

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Frobert, C. (1996). Compression asymétrique de séquences stéréoscopiques par compensation de mouvement/disparité et modélisation fractale [Master's thesis, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/8977/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/8977/>
PolyPublie URL:

Directeurs de recherche: Paul Cohen
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

COMPRESSION ASYMÉTRIQUE DE SÉQUENCES STÉRÉOSCOPIQUES PAR
COMPENSATION DE MOUVEMENT/DISPARITÉ ET MODÉLISATION
FRACTALE

Christine FROBERT

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLOME DE MAÎTRE ÈS SCIENCES APPLIQUÉES (M.Sc.A.)
(GÉNIE ÉLECTRIQUE)

décembre 1996

© droits réservés de Christine FROBERT 1996.



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

Our file *Notre référence*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-26471-8

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Ce mémoire intitulé:

COMPRESSION ASYMÉTRIQUE DE SÉQUENCES
STÉRÉOSCOPIQUES PAR COMPENSATION DE
MOUVEMENT/DISPARITÉ ET MODÉLISATION FRACTALE

présenté par: Christine FROBERT

en vue de l'obtention du diplôme de: Maître ès science appliquées

a été dûment accepté par le jury d'examen constitué de:

Hervé Jean-Yves, Ph.D., président

Cohen Paul, Ph.D., membre et directeur de recherche

Houle Jean-Louis, Ph.D., membre

A mon cher grand-oncle Louis Largy.

Remerciements

Je tiens tout d'abord à remercier Noranda et le CRSNG pour leur soutien financier. Je tiens aussi à remercier le professeur Paul Cohen, mon directeur de recherche, pour son aide et ses judicieux conseils tout au cours de ce projet de recherche.

Je remercie également les autres membres du Groupe de Recherche en Perception et Robotique de l'École Polytechnique de Montréal, étudiants, assistants de recherche ou professeurs pour leur soutien tout au long de mes études de maîtrise.

Enfin, je remercie mes parents, Jacqueline et Jean-Pierre ainsi que mon frère Laurent pour leur amour et leur confiance de tous les instants.

Résumé

Dans ce mémoire, nous présentons un système original de compression de séquences d'images stéréoscopiques basé sur un codage multi-niveaux hiérarchique. En outre, le codec utilise la théorie de la neutralisation du système visuel humain en codant les images stéréoscopiques de manière asymétrique augmentant ainsi le taux de compression. Selon cette théorie empirique, lorsque les deux yeux reçoivent simultanément deux images stéréoscopiques de qualité inégale, le cerveau demeure capable (dans certaines limites) de fusionner ces deux images en une image tri-dimensionnelle de haute qualité. Dans une stratégie de codage, cette propriété se traduit par le codage des deux séquences homologues à des résolutions différentes. La séquence à basse résolution, notée L , est filtrée passe-bas et sous-échantillonnée par une pyramide gaussienne, ce qui permet de réduire sa résolution de moitié selon les deux dimensions. Ensuite, le codage de la séquence à haute résolution, notée H , suit les étapes suivantes. Chaque trame H_t est partitionnée en blocs images I_t^i non superposés représentant le contenu de l'image à la position i et au temps t . Une première tentative de compensation du mouvement sans mouvement est effectuée en comparant I_t^i avec \hat{I}_{t-1}^i , le bloc situé à la

même position dans la trame reconstruite précédente. Seuls les blocs I_t^i actifs (pour lesquels l'erreur entre I_t^i et \hat{I}_{t-1}^i dépasse un seuil donné) sont codés. Le système de codage hybride consiste alors à approximer chaque bloc I_t^i actif par un bloc domaine \hat{D}_{t-1}^j transformé provenant de la trame reconstruite précédente. D'abord, une tentative de compensation du mouvement utilisant un modèle translationnel est menée pour coder I_t^i . Si l'approximation résultante n'est pas satisfaisante, I_t^i est codé en utilisant une modélisation fractale. Le codage fractal global consiste à approximer I_t^i par un bloc domaine \hat{D}_{t-1}^j soumis à des transformations non linéaires de types spatiales, massiques et géométriques. Le traitement est effectué dans le domaine fréquentiel afin d'accélérer l'étape de codage par la construction d'une représentation invariante aux transformations fractales contrôlée par une étude de compatibilité de signes. Bien qu'introduisant des calculs supplémentaires, cette étude n'augmente pas le temps de calcul au codeur car elle permet de réduire le nombre de blocs domaines candidats à considérer ce qui économise des calculs inutiles. Si l'approximation résultante n'est toujours pas suffisante, le bloc I_t^i est segmenté en quatre sous-blocs de même taille et le processus complet de codage est répété pour chaque sous-bloc. Cette structure permet de coder de manière hiérarchique et adaptative (avec des taux de compression variables) des régions statiques, en mouvement et complexes (par exemple correspondant à des zones d'occlusion) de la scène. En outre, le processus de codage est mené de manière purement inter-trame c'est-à-dire qu'une image est codée à partir de l'information disponible dans l'image précédente. Cette propriété de causalité permet

d'avoir un décodage non itératif.

La version filtrée de la séquence à basse résolution L^{fil} peut suivre les mêmes étapes de codage que celles de la séquence H ou exploiter explicitement la géométrie stéréoscopique. Dans ce cas, le module de compensation du mouvement est remplacé par un module de compensation de disparité qui consiste à approximer le bloc I_t^l par un bloc \hat{D}_t^l recherché à la même itération dans l'image homologue H_t . Cette stratégie permet d'augmenter la compression mais rend le codage/décodage de la séquence L^{fil} dépendant de celui de H .

Abstract

In this thesis, we present a new block-based method for stereoscopic video compression based on a 3-step hierarchical encoding scheme. The binocular rivalry theory is integrated in the conception of the asymmetric encoding of the stereoscopic images to increase the compression performance. This empirical theory states that, provided with two stereoscopic images with different resolutions, the brain remains able (with respect to certain constraints) to fuse them into a 3-D high-quality image. In a coding scheme, this results into two sequences of different quality. The low-quality sequence, L , is low-pass filtered and subsampled by a gaussian pyramid which allows to reduce its resolution by a factor 2 along each dimension. The high-quality sequence, H , is encoded according to the following steps. Each frame H_t is partitioned into non overlapping image blocks I_t^i representing image data at a specific location i over time t . A crude form of motion compensation without any motion is first done by comparing I_t^i with \hat{I}_{t-1}^i , the block located at the same position in the previous reconstructed frame. Only active I_t^i (for which the root mean square difference between I_t^i and \hat{I}_{t-1}^i is above a given threshold) are coded. The hybrid coding scheme then consists in

approximating each active I_t^i by a transformed domain block \hat{D}_{t-1}^j from the previous reconstructed frame. First, an attempt is made to encode I_t^i by translational motion compensation. If the resulting approximation is not satisfying, I_t^i is fractal-encoded. The global fractal processing consists in approximating I_t^i by a domain block \hat{D}_{t-1}^j transformed by non linear (spatial, massic and geometric) transformations. The process is made in the frequency domain to accelerate the encoding step by building invariant to fractal transformation features controlled by a study of compatibility of signs. Although it introduces new calculations, this study does not increase the time coding because it allows the rejection of candidate domain blocks. If the resulting approximation is again not satisfying, I_t^i is quad-tree partitionned and the entire process is repeated for the four corresponding sub-blocks. This structure allows to encode hierarchically and adaptively (with varying compression ratios) static, moving and complex (for instance occluded) regions of the image. Besides, the encoding process is entirely inter-frame which means that a frame is encoded with the informations available in the previous frame. This causality allows a non-iterative decoding.

The filtered version of the low-quality sequence L^{fil} may follow the same scheme as H or exploit explicetely the stereoscopic geometry. In that case, the motion compensation module is replaced by a disparity compensation one which consists of approximating I_t^i by a domain block \hat{D}_t^j located in the reconstructed homolog frame at same iteration \hat{H}_t . This strategy increases the compression ratio but leads to a coding/decoding of L^{fil} depending upon that of H .

Table des matières

Dédicace	iv
Remerciements	v
Résumé	vi
Abstract	ix
Table des matières	xi
Liste des figures	xii
Liste des tableaux	xiii
Liste des annexes	xiv
Liste des notations	xv
Introduction	1
1 Revue bibliographique	7

1.1	Compensation du mouvement	8
1.1.1	Principe du codage par compensation du mouvement	8
1.1.2	Modélisation du mouvement	9
1.1.3	Méthode de base à modèle translationnel	11
1.2	Codage fractal	13
1.2.1	Théorie des transformées itératives	14
1.2.2	Structure et construction des codes par fractales	15
1.2.3	Méthode de codage de base	16
1.3	Stéréoscopie et théorie de la neutralisation	18
1.3.1	Définitions et concepts	18
1.3.2	Phénomène de rivalité binoculaire	20
1.3.3	Utilisation de la théorie de la neutralisation pour le codage d'images stéréoscopiques	21
1.4	Discussion	22
2	Présentation générale du système	25
2.1	Description des critères	26
2.2	Codage de la séquence à haute résolution H	27
2.3	Codage de la séquence à faible résolution L	32
3	Description de la méthode de codage	35
3.1	Codage de la séquence à haute résolution H	36

3.1.1	Module d'estimation des blocs actifs	37
3.1.2	Module de compensation du mouvement	40
3.1.3	Module de codage fractal	44
3.1.4	Module de segmentation en arbre quaternaire	78
3.2	Codage de la séquence à faible résolution L	85
3.2.1	Pyramide gaussienne	86
3.2.2	Module de compensation de disparité	87
3.2.3	Étude de l'influence des seuils et paramètres	89
3.3	Éléments constitutifs du décodeur	91
3.4	Bilan	94
3.5	Implantation	96
3.5.1	Notations utilisées	96
3.5.2	Décodabilité du système	96
4	Résultats expérimentaux	107
4.1	Codage de séquences d'images monoculaires	107
4.1.1	Méthodologie	107
4.1.2	Justification d'un mode de codage hybride	109
4.1.3	Comparaison de notre système avec <i>MPEG - 1</i>	117
4.2	Codage de séquences d'images stéréoscopiques	123
4.2.1	Dispositif d'affichage	123
4.2.2	Comparaison de la performance de diverses variantes	123

Conclusion	127
Bibliographie	133
ANNEXES	151

Liste des figures

1.1	Système typique de codage vidéo par compensation du mouvement	9
1.2	Processus d'estimation du mouvement	12
1.3	Illustration du point fixe, extrait de Fisher (1995)	14
1.4	Principe du codage fractal par bloc	18
1.5	La rivalité binoculaire ne détruit pas la stéréopsie	20
2.1	Schéma bloc du codeur	26
3.1	Schéma bloc du codeur	36
3.2	Module de compensation du mouvement	42
3.3	Influence du seuil de succès du module de compensation du mouvement Γ_{mut} sur le rapport entre le taux de compression et l'erreur quadratique moyenne	43
3.4	Influence de la longueur de la spirale de recherche λ_{mut} sur le rapport entre le taux de compression et l'erreur quadratique moyenne	45
3.5	Module de codage fractal	46

3.6	Transformations Ψ dans le domaine spatial et fréquentiel	51
3.7	Représentation canonique d'un bloc	60
3.8	Exemple pathologique de la représentation invariante	62
3.9	Organigramme de l'étude de compatibilité des blocs	65
3.10	Exemple d'étude de compatibilité de signes	69
3.11	Influence de la longueur de la spirale de recherche λ_{frac} sur le rapport entre le taux de compression et l'erreur quadratique moyenne	71
3.12	Influence de la quantification du facteur d'échelle a sur le rapport entre le taux de compression et l'erreur quadratique moyenne	73
3.13	Influence de la quantification du facteur de décalage b sur le rapport entre le taux de compression et l'erreur quadratique moyenne	75
3.14	Influence des paramètres d'étude de compatibilité des signes Γ_{prep} et Δ_{prep}^{max} sur le rapport entre le taux de compression et l'erreur quadratique moyenne	77
3.15	Principe de segmentation en arbre quaternaire	79
3.16	Influence de la segmentation en arbre quaternaire des blocs actifs sur le rapport entre le taux de compression et l'erreur quadratique moyenne	80
3.17	Influence de la segmentation en arbre quaternaire des blocs actifs non estimables par compensation du mouvement sur le rapport entre le taux de compression et l'erreur quadratique moyenne 1	82

3.18	Influence de la segmentation en arbre quaternaire des blocs actifs non estimables par compensation du mouvement sur le rapport entre le taux de compression et l'erreur quadratique moyenne 2	83
3.19	Illustration de la théorie de la neutralisation	86
3.20	Module de compensation de disparité	89
3.21	Influence du seuil Γ_{disp} sur le rapport entre le taux de compression et l'erreur quadratique moyenne	90
3.22	Influence de $\lambda_{disp,x}$ sur le rapport entre le taux de compression et l'erreur quadratique moyenne	91
3.23	Schéma explicatif du décodeur stéréoscopique	93
3.24	Schéma bloc du codec	95
3.25	Flux binaires transmis selon le mode de transmission	99
3.26	Comparaison des trois modes de transmission	105
4.1	Rapport $PSNR$ obtenu selon les trois modes de codage	111
4.2	Taux de compression τ obtenu selon les trois modes de codage	112
4.3	Produit $\tau \times PSNR$ obtenu selon les trois modes de codage	113
4.4	Rapport $PSNR$ obtenu avec notre système et $MPEG - 1$	119
A.1	Pyramide gaussienne	170
B.1	Première image de chacune des séquences originales sur lesquelles les tests ont été menés	176

B.2	Justification du mode hybride de codage	177
B.3	Justification du mode hybride de codage, suite	178
B.4	Comparaison du mode de codage hybride avec <i>MPEG</i> – 1	179
B.5	Comparaison du mode de codage hybride avec <i>MPEG</i> – 1, suite . . .	180
B.6	Comparaison entre les variantes de codage de la séquence à faible résolution	181
C.1	Compatibilité des signes pour 3 classes différentes	186
C.2	Exemple d'étude de compatibilité de signes	191
D.1	Influence du seuil de succès du module de compensation du mouvement Γ_{mut} sur le taux de compression	193
D.2	Influence du seuil de succès du module de compensation du mouvement Γ_{mut} sur l'erreur quadratique moyenne	194
D.3	Influence de la longueur de la spirale de recherche λ_{mut} sur le taux de compression	195
D.4	Influence de la longueur de la spirale de recherche λ_{mut} sur l'erreur quadratique moyenne	196
D.5	Influence de la longueur de la spirale de recherche λ_{frac} sur le taux de compression	198
D.6	Influence de la longueur de la spirale de recherche λ_{frac} sur l'erreur quadratique moyenne	199

D.7	Influence de la quantification du facteur d'échelle a sur le taux de compression	200
D.8	Influence de la quantification du facteur d'échelle a sur l'erreur quadratique moyenne	201
D.9	Influence de la quantification du facteur de décalage b sur le taux de compression	202
D.10	Influence de la quantification du facteur de décalage b sur l'erreur quadratique moyenne	203
D.11	Influence du nombre maximum de termes prépondérants Δ_{prep}^{max} et du seuil de prépondérance Γ_{prep} sur le taux de compression	204
D.12	Influence du nombre maximum de termes prépondérants Δ_{prep}^{max} et du seuil de prépondérance Γ_{prep} sur l'erreur quadratique moyenne	205
D.13	Influence du seuil de succès du module de compensation de disparité Γ_{disp} sur le taux de compression	206
D.14	Influence du seuil de succès du module de compensation de disparité Γ_{disp} sur l'erreur quadratique moyenne	207
D.15	Influence de la longueur du domaine de recherche horizontal $\lambda_{disp,x}$ sur le taux de compression	207
D.16	Influence de la longueur du domaine de recherche horizontal $\lambda_{disp,x}$ sur l'erreur quadratique moyenne	208

Liste des tableaux

3.1	Classes des coefficients TCD	58
3.2	Ensemble des paramètres du codeur	94
3.3	Notations utilisées	97
3.4	Catégories de blocs générés et nombre de bits moyens d'entête avec chaque mode	100
4.1	Echelle de visibilité des dégradations	109
4.2	Résultats subjectifs internes sur la séquence "Rubic"	116
4.3	Résultats subjectifs internes sur la séquence "Taxi"	116
4.4	Résultats subjectifs internes sur la séquence "NASA"	116
4.5	Résultats subjectifs internes sur la séquence "Papillon"	117
4.6	Résultats subjectifs internes sur la séquence "Tunnel"	117
4.7	Liste des tests subjectifs effectués sur chacun des sujets	120
4.8	Résultats subjectifs externes sur la séquence "Rubic"	122
4.9	Résultats subjectifs externes sur la séquence "Taxi"	123

4.10 Résultats subjectifs externes sur la séquence “NASA”	123
4.11 Résultats subjectifs externes sur la séquence “Papillon”	123
4.12 Résultats subjectifs externes sur la séquence “Tunnel”	124
4.13 Résultats subjectifs stéréoscopiques sur la séquence “Papillon”	126
4.14 Résultats subjectifs stéréoscopiques sur la séquence “Tunnel”	126
B.1 Description des séquences types sur lesquelles les tests ont été menés	175

Liste des annexes

A	Revue Bibliographique: Compléments	151
A.1	Méthodes de codage par compensation du mouvement	152
A.1.1	Autres méthodes de compensation du mouvement par blocs	152
A.1.2	Compensation du mouvement par régions	155
A.1.3	Compensation dense	157
A.2	Codage fractal	157
A.2.1	Compression d'images fixes	157
A.2.2	Extension au codage de séquences vidéo	163
A.3	Stéréoscopie et théorie de la neutralisation	168
A.3.1	Méthodes de codage utilisées	168
A.3.2	Techniques d'expérimentations	171
B	Séquences d'images tests	174
B.1	Description des séquences	175
B.2	Première image de chaque séquence originale	175

B.3	Dernière image reconstruite de chaque séquence	177
C	Construction d'une représentation invariante: Étude détaillée des	
	compatibilités de signe	182
D	Influence des différents paramètres: Courbes complémentaires . .	192
D.1	Module de compensation du mouvement	192
D.2	Module de codage fractal	197
D.3	Module de codage stéréoscopique	206

Liste des notations

- H_t : image de la séquence stéréoscopique à haute résolution H au temps t ;
- L_t : image de la séquence stéréoscopique à basse résolution L au temps t ;
- MAX_X : largeur de l'image;
- MAX_Y : hauteur de l'image;
- $B_t(x, y)$: intensité lumineuse du pixel (x, y) du bloc B au temps t ;
- $\hat{B}(x, y)$: intensité lumineuse du pixel (x, y) du bloc B reconstruit;
- $B^{fil}(x, y)$: intensité lumineuse du pixel (x, y) du bloc B de l'image filtrée par une pyramide gaussienne;
- $B^{int}(x, y)$: intensité lumineuse du pixel (x, y) du bloc B de l'image reconstruite et interpolée;
- \mathcal{B} : bloc B transformé par TCD ;
- \mathcal{B}^{nor} : bloc obtenu après application de l'invariance massique sur le bloc \mathcal{B} ;

- \mathcal{B}^{trans} : bloc obtenu après application de l'invariance géométrique de transposition sur le bloc \mathcal{B}^{nor} ;
- \mathcal{B}^{can} : bloc canonique obtenu après application de l'invariance géométrique de masque sur le bloc \mathcal{B}^{trans} ;
- $\Gamma_{act,N,t}$: seuil de succès du module d'estimation des blocs actifs de taille $N \times N$ de l'image au temps t ;
- \mathcal{L}_t : pourcentage de blocs actifs considérés dans l'image au temps t ;
- Γ_{mvt} : seuil de succès du module de compensation du mouvement;
- λ_{mvt} : longueur de la spirale du module de compensation du mouvement;
- λ_{frac} : longueur de la spirale du module de codage fractal;
- $\lambda_{disp,x}$: longueur de la composante x du vecteur de disparité du module de compensation de disparité;
- $\lambda_{disp,y}$: longueur de la composante y du vecteur de disparité du module de compensation de disparité;
- τ : taux de compression obtenu pour le codage de l'image courante;
- $MPEG - 1$: "Motion Picture Expert Group" 1-ère version;
- $JPEG$: "Joint Picture Expert Group";
- TCD : Transformation en cosinus discrète;

- *EQM*: Erreur quadratique moyenne:
- *MDA*: Erreur moyenne des différences absolues:
- *PSNR*: Rapport signal sur bruit maximum:

Introduction

L'imagerie stéréoscopique constitue un domaine de recherche actuellement en plein essor. En effet, la perception de la profondeur assurée par la fusion de paires d'images stéréoscopiques constitue une information précieuse pour interpréter de manière précise et instantanée une scène et interagir avec ses différents éléments. L'ajout de cette troisième dimension impossible avec des séquences monoculaires joue un rôle primordial dans des domaines aussi nombreux que variés comme la médecine (opérations chirurgicales assistées par l'imagerie), l'industrie des divertissements (jeux vidéos) ou la télérobotique. Le problème majeur réside dans le fait que la quantité d'information existant dans une séquence d'images stéréoscopiques est gigantesque, et un effort tout particulier doit être mis à la réduction de cette information avant transmission. Dans cette optique, le but de la compression avec perte est de représenter des images ou des séquences d'images sous la forme la plus compacte possible tout en assurant que les dégradations introduites ne seront pas gênantes visuellement pour l'application considérée. Cette compression est généralement possible car une séquence d'images stéréoscopiques contient de l'information répétitive appelée redondance (Bhaskaran

et Konstantinides 1995).

Une première source de redondance réside dans la structure intra-image. En effet, une image contient un ou plusieurs objets et un pixel correspondant à un objet de la scène voit très souvent sa luminance corrélée avec celle d'un autre pixel du même objet. L'exploitation de cette redondance dite spatiale consiste à prédire la luminance d'un pixel à partir de celle de ses voisins. La deuxième source de redondance présente dans une séquence d'images provient du fait qu'au cours du temps, on retrouve les mêmes objets ou parties d'objets à la même position ou déplacés dans une autre région de l'image. L'exploitation de cette redondance dite temporelle va alors consister à estimer et à compenser le déplacement des pixels entre plusieurs images successives d'une séquence. Au niveau d'une seule image ou de plusieurs images successives, il existe aussi une redondance, dite fractale, qui consiste en la répétition d'information à différentes échelles. Enfin, la dernière source de redondance disponible dans des séquences d'images stéréoscopiques est la répétition d'information entre deux images homologues. En effet, deux images stéréoscopiques correspondent à deux points de vue très légèrement différents d'une même scène. Ainsi, un objet visible dans une image a de grandes chances d'apparaître, même partiellement occulté, dans l'image homologue. L'exploitation de la redondance stéréoscopique peut consister à effectuer la mise en correspondance des deux images en évaluant la différence de position—appelée disparité—d'un pixel correspondant à la projection du même point physique de la scène dans les deux images.

Outre la répétition d'information, de la compression supplémentaire peut être obtenue en exploitant les limites des capacités perceptuelles du système visuel humain (Arditi 1986). Ainsi, certaines informations n'ont pas besoin d'être transmises puisque, de toute façon, elles ne seront pas discernées en raison de la résolution spatiale ou temporelle limitée du système visuel humain. Pour ce qui est de la vision binoculaire humaine, d'autres caractéristiques peuvent être exploitées comme le fait que la précision de la perception de la profondeur varie avec la distance entre l'observateur et l'objet fusionné. En outre, il est communément reconnu que les individus possédant deux yeux d'acuité visuelle inégale perçoivent le monde environnant avec la qualité de leur meilleur œil. Dans une stratégie de compression, ceci pourrait se traduire par une réduction volontaire de la résolution d'une des deux séquences stéréoscopiques. Certains critères doivent être respectés lors de la conception d'un système de codage de séquences d'images. Ainsi, des compromis doivent être faits entre plusieurs contraintes contradictoires: la qualité visuelle de la séquence reconstruite, le taux de compression obtenu, le délai de codage et la complexité d'implantation du système. La prévalence d'un critère par rapport à un autre dépend de l'application considérée. En outre, dans le cas de la stéréoscopie, il faut de plus tenir compte de la qualité de l'image fusionnée, généralement évaluée par des critères subjectifs.

En ce qui concerne la compression d'images et de séquences vidéos, la plupart des normes établies comme *JPEG*, *MPEG* et *H-261* sont basées sur un codage par compensation du mouvement et exploitent les redondances d'information spatiale et

temporelle. Nous allons voir que ces méthodes présentent l'avantage d'être très rapides mais ont une performance qui dépend de la complexité du mouvement dans la séquence. En fait, elles demeurent limitées pour ce qui est du taux de compression et de la qualité visuelle obtenus. De ce fait, parallèlement à ces normes, un grand effort de recherche est aujourd'hui consacré à des méthodes alternatives. En particulier, après la popularité fulgurante des ondelettes, c'est actuellement la théorie des transformées itératives appelée aussi théorie des fractales qui jouit d'un intérêt grandissant. Nous allons voir qu'un codage basé sur l'exploitation de la redondance fractale est libéré de certains des défauts des méthodes basées sur la compensation du mouvement. Par contre, il présente généralement un temps de calcul plus long. Pour ce qui est du codage d'images ou de séquences stéréoscopiques, deux grandes théories s'affrontent conduisant à des stratégies de codage opposées (Labonté et Laganière 1996). La *théorie de la fusion* de la perception de la profondeur suppose que les informations fournies par les deux yeux se combinent résultant en un modèle mental volumétrique du monde environnant. Cette théorie conduit donc à coder les deux images stéréoscopiques de manière symétrique ce qui implique la création d'une représentation intermédiaire intégrant l'information des deux images d'une paire stéréoscopique. Par opposition, la *théorie de la neutralisation* considère qu'une des deux images de la paire stéréoscopique doit retenir les détails de la scène tandis que la seconde image peut représenter seulement l'information de disparité. Cette théorie inspire donc les approches de codage dites asymétriques, où les deux images d'une

paire stéréoscopique sont traitées différemment, généralement en utilisant une image comme une référence pour la prédiction de l'autre. Une des deux images est ainsi codée à une résolution plus faible que l'autre.

L'objectif de ce mémoire est de développer un système de codage asymétrique de séquences d'images stéréoscopiques respectant les contraintes qui suivent. Le taux de compression global obtenu pour les deux séquences doit se rapprocher de celui que l'on obtiendrait en codant une séquence monoculaire. Pour cela, nous utilisons la théorie de la neutralisation en compressant très sévèrement une des deux séquences. La qualité de la séquence à haute résolution doit être comparable ou supérieure à celle obtenue en utilisant une méthode basée sur la norme *MPEG - 1* pour des taux de compression comparables. Enfin, nous privilégions la simplicité algorithmique et la rapidité d'exécution du codec dans un but futur d'implantation matérielle en temps réel. Ces dernières considérations nous dirigent vers une méthode de codage hybride par compensation de mouvement/disparité et par codage fractal basée sur un partitionnement de l'image en blocs de taille variable.

Le mémoire est organisé comme suit. Dans le chapitre 1, nous présentons la revue bibliographique sur laquelle s'est appuyé ce travail. Les détails de cette bibliographie sont présentés dans l'annexe A. D'abord, nous décrivons succinctement le principe et les principales méthodes de codage basées sur la compensation du mouvement en insistant tout particulièrement sur les méthodes basées-blocs. Puis, nous détaillons le concept du codage fractal ainsi qu'une synthèse des principales méthodes de com-

pression d'images fixes et de séquences vidéos. Ensuite, nous présentons les bases de la vision binoculaire humaine soutenant la théorie de la neutralisation. Enfin, nous décrivons les principales techniques de codage asymétrique de paires d'images stéréoscopiques intégrant cette théorie. Le chapitre 2 justifie très succinctement les choix que nous avons faits à partir des contraintes de notre cahier des charges pour orienter la conception de notre méthode. Le chapitre 3 constitue le coeur de ce mémoire. Il décrit la méthode de codage que nous avons choisi en décomposant ses différents modules et en étudiant minutieusement tous les paramètres qu'il utilise. L'ensemble des résultats expérimentaux avec des comparaisons de performance internes et avec *MPEG-1* à partir de tests d'évaluation de qualité objectifs et subjectifs est regroupé dans le chapitre 4. Enfin, une conclusion vient résumer l'ensemble du travail effectué en décrivant les caractéristiques de notre système ainsi que les résultats obtenus et propose des éventuels développements en vue d'améliorer notre méthode.

Chapitre 1

Revue bibliographique

Ce chapitre est divisé en trois sections. Les deux premières sections décrivent les concepts de deux approches de codage que nous allons utiliser pour représenter le mouvement dans une séquence d'images: la compensation du mouvement et la modélisation fractale. La troisième partie concerne plus spécifiquement le codage de paires d'images stéréoscopiques. Elle présente la théorie de la neutralisation relative à la vision stéréoscopique humaine. Une description succincte des méthodes relatives à chacune des stratégies de codage se trouve dans l'annexe A. Enfin, la section 1.4 effectue une analyse critique des approches par compensation du mouvement et codage fractal existantes.

1.1 Compensation du mouvement

Le codage par compensation du mouvement est la méthode la plus populaire pour la compression des séquences d'images. Elle permet de réduire l'information essentielle dans des images en éliminant une partie de la redondance temporelle. La prédiction temporelle avec compensation du mouvement consiste à obtenir un estimé de l'image courante à partir d'une mesure des mouvements des intensités lumineuses¹ entre celle-ci et les images qui la précèdent ou la suivent. Ces mouvements sont obtenus par établissement de correspondances entre ces images. Pour ceci, il faut se baser sur le mouvement projeté dans ces images. Il existe plusieurs modèles de mouvement utilisés à cette fin que nous décrivons dans la section 1.1.2. Dans la section 1.1.1, nous exposons le principe du codage par compensation du mouvement. Les systèmes de codage par compensation du mouvement existant peuvent être classés dans plusieurs catégories et nous présentons la méthode la plus couramment utilisée dans la section 1.1.3.

1.1.1 Principe du codage par compensation du mouvement

Le schéma générique d'un système de codage vidéo par compensation du mouvement est illustré à la figure 1.1. Dans cette figure, la luminance d'un pixel (x, y) de l'image I_t est prédite à partir de $\hat{I}_{t-1}(x', y')$ sa luminance reconstruite à l'image

¹la contrainte du gradient suppose que la luminance d'un pixel ne varie pas entre des images successives

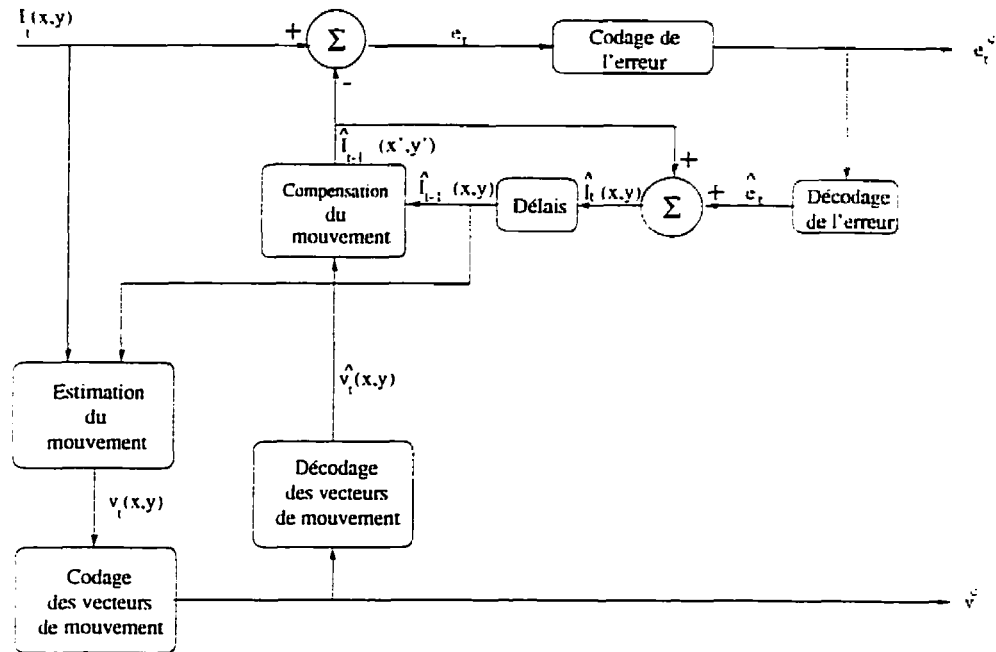


Figure 1.1 : Système typique de codage vidéo par compensation du mouvement

précédente et déplacée à la position (x', y') . Pour cela, il a fallu effectuer l'estimation du mouvement du pixel (x, y) en trouvant, par une mise en correspondance dans \hat{I}_{t-1} , le meilleur pixel (x', y') pouvant l'approximer. Le déplacement estimé, appelé vecteur de mouvement est noté $v(x, y)$ et codé par quantification.

Ensuite, l'erreur de prédiction e_t entre le pixel original et le pixel estimé est codée.

1.1.2 Modélisation du mouvement

Pour effectuer l'estimation du mouvement, il faut mettre en correspondance un pixel (x, y) et le même pixel déplacé à la position (x', y') . Ce déplacement est soumis à différentes contraintes selon les hypothèses faites quant au vrai mouvement dans la scène. Ces hypothèses mènent à l'établissement d'une modélisation plus ou moins

simplificatrice du mouvement qui permet de déterminer les composantes du vecteur de déplacement $v(x, y)$.

Ainsi, les travaux basés sur un modèle de mouvement supposent qu'une région de l'image provient d'un objet de la scène qui subit un mouvement rigide bien descriptible par le modèle. Plusieurs modèles ont été proposés. Quand le nombre de paramètres définissant le modèle augmente, il devient mieux représentatif du champ de mouvement² mais il devient aussi plus compliqué à estimer et plus coûteux à transmettre. Les modèles les plus souvent utilisés dans la littérature peuvent être trouvés dans (Diab et Cohen 1996). Le modèle le plus couramment utilisé dans les systèmes de codage actuels est le suivant:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix}.$$

Ce modèle à deux paramètres dit translationnel est employé dans les méthodes de compensation du mouvement par blocs et associe un vecteur de mouvement (u, v) à chaque bloc $I(x, y)$. Ce modèle revient à ne prendre en compte que des déplacements translationnels fronto-parallèles par rapport au plan de la caméra. Ceci restreint très sévèrement les mouvements physiquement possibles, mais il se distingue par sa simplicité et permet une implantation matérielle en temps réel.

²Le champ de mouvement représente le vrai mouvement tri-dimensionnel des objets de la scène.

1.1.3 Méthode de base à modèle translationnel

C'est une méthode de compensation du mouvement par blocs de taille fixe (en général 16×16). L'erreur de prédiction définie dans la section précédente vaut:

$$e_t(x, y) = I_t(x, y) - \hat{I}_{t-1}(x + u, y + v),$$

où $I_t(x, y)$ est la valeur du pixel en (x, y) dans l'image au temps t et $\hat{I}_{t-1}(x + u, y + v)$ est la valeur du pixel correspondant en $(x + u, y + v)$ dans l'image reconstruite au temps $t - 1$. Le vecteur (u, v) représente le mouvement relatif translationnel d'un bloc d'une image à l'autre et est déterminé lors de l'étape d'estimation du mouvement.

La technique d'estimation du mouvement est illustrée à la figure 1.2. Soit un bloc à coder de position (x, y) (définie par le coin supérieur gauche du bloc) dans l'image au temps t . Ce bloc va être mis en correspondance avec un ensemble de blocs situés dans l'image reconstruite au temps $t - 1$ dans une région de recherche centrée autour de la position (x, y) . Le vecteur de mouvement (u, v) représente la position relative du meilleur bloc trouvé dans la région de recherche.

Il existe plusieurs algorithmes pour déterminer les vecteurs (u, v) (Bhaskaran et Konstantinides 1995). Le plus simple à implanter mais aussi le plus long en temps de calcul est celui basé sur la méthode de recherche exhaustive. Cette méthode consiste à parcourir uniformément selon un pas constant les deux dimensions x et y de la région de recherche pour trouver le meilleur bloc déplacé et donc le vecteur de mouvement

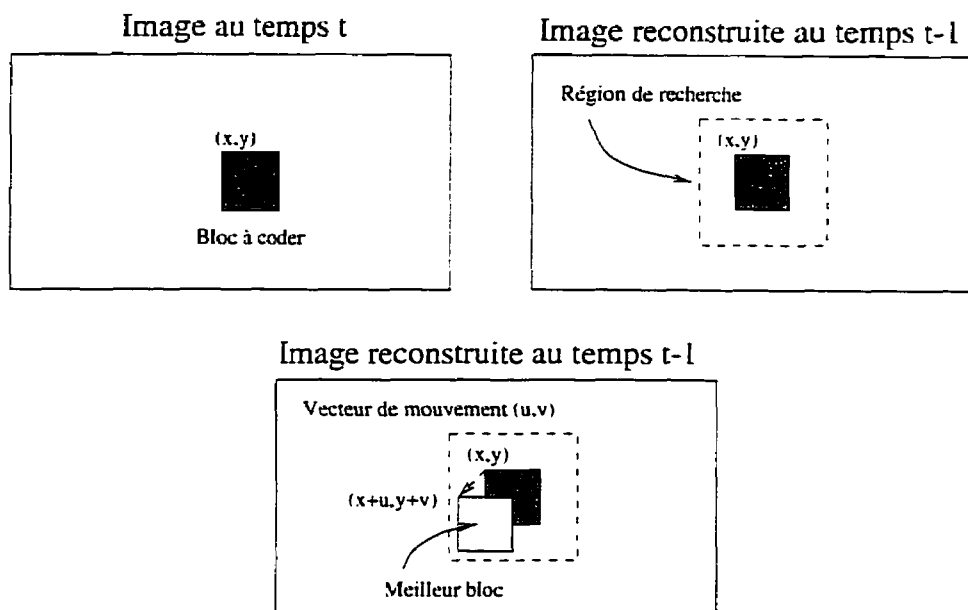


Figure 1.2 : Processus d'estimation du mouvement

(u, v) .

Quel que soit l'algorithme de recherche utilisé, le meilleur bloc sera celui qui minimise un critère de distorsion donné entre le bloc codé et le bloc estimé. Il existe principalement deux critères de distorsion : l'erreur de la moyenne des différences absolues (*MDA*) et l'erreur quadratique moyenne (*EQM*).

La norme MPEG (ISO 1990, Gall 1991, Bhaskaran et Konstantinides 1995) utilise cette méthode de codage, mais de manière plus élaborée car elle distingue plusieurs types d'images codées de manière intra-trame (images de mise à jour sans compensation du mouvement) ou inter-trame.

1.2 Codage fractal

La théorie par modélisation fractale exploite la redondance d'information de type fractale existant dans l'image. Cette théorie prend naissance avec les travaux de Barnsley en 1987. En 1989, Jacquin (1992) met au point le premier codeur complètement automatisé basé-bloc pour les images fixes monochromes utilisant le concept de transformées itératives. Depuis, les variantes et les améliorations se sont multipliées et se sont étendues au codage de séquences vidéos.

Un objet (comme une image) est considéré comme fractal s'il peut être défini par la simple connaissance d'une transformation mathématique particulière T définie ci-dessous. Lorsque T est appliquée récursivement à un objet de départ arbitraire et les résultats successifs de l'objet transformé, elle converge vers l'objet fractal, dit point fixe de T . Il existe donc une relation univoque entre T et son point fixe.

La figure 1.3 empruntée à Fisher (1995) montre qu'une même transformation, appliquée à des images différentes, aboutit en un petit nombre d'itérations (3) vers le même point fixe, en l'occurrence le triangle de Serpinski.

T doit obéir à certaines propriétés mathématiques pour assurer la convergence globale de la transformation. L'aspect mathématique du codage fractal est survolé dans la section 1.2.1 que nous adaptons aux objets "images" dans la section 1.2.2. Enfin, la méthode de codage de base utilisant la théorie des transformées itératives est présentée dans la section 1.2.3.

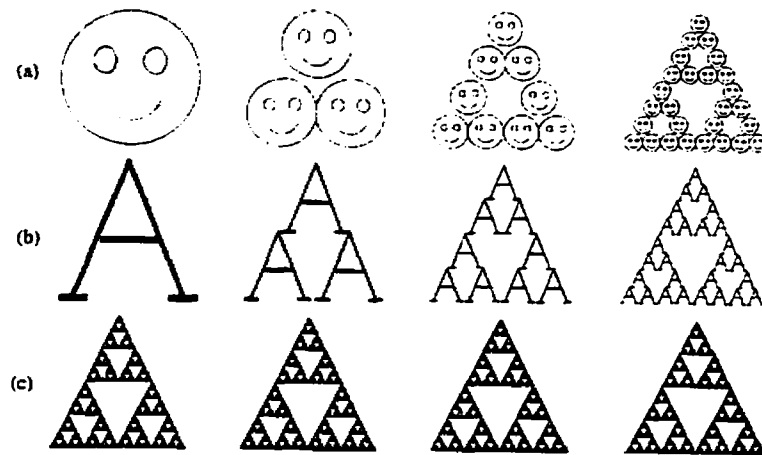


Figure 1.3 : Illustration du point fixe, extrait de Fisher (1995)

1.2.1 Théorie des transformées itératives

Soit (M, d) un espace métrique pour les images digitales, où d représente une mesure de distorsion (comme l'erreur quadratique moyenne) et soit f_{orig} l'image originale que nous voulons coder. Le problème inverse de la théorie des transformations itératives consiste à construire T une transformation d'image contractive, définie de l'espace (M, d) vers lui-même, pour laquelle f_{orig} constitue un point fixe approximatif. La transformation T doit remplir les conditions suivantes :

$$\exists s < 1 : \forall f, g \in M, \quad d(T(f), T(g)) \leq s d(f, g),$$

$$\exists \epsilon : \quad d(f_{orig}, T(f_{orig})) \leq \epsilon.$$

La première relation représente la contrainte de contractivité de la transformation T . Cette contrainte est nécessaire pour assurer la convergence de la suite $T^n(f_0)$. La

deuxième relation formule l'approximation du point fixe que constitue f_{orig} .

Si ces deux conditions sont satisfaites, la suite $f_n = T^n(f_0)_{n \geq 0}$ où f_0 est une image initiale arbitraire, converge vers l'image originale f_{orig} . Dans l'espace des images quantifiées, la séquence converge exactement vers une image stable, qui, résultant d'une construction itérative, est dite fractale.

1.2.2 Structure et construction des codes par fractales

En pratique, la similarité propre rencontrée dans les images naturelles diffère de celle existant dans les objets fractals. Au lieu d'avoir une image formée de copies d'elle-même (dans son ensemble), une image est formée de copies de morceaux adéquatement transformés d'elle-même. Par conséquent, la classe des transformations contractives à considérer pour exploiter cette similarité propre par morceaux est définie par blocs.

Soit $I_i, 0 \leq i < N$ un partitionnement du support de l'image—avec ou sans superposition—en N cellules³. Nous avons alors :

$$f = \sum_{i=0}^{N-1} (f|_{I_i}),$$

$$\forall f \in M, T(f) = \sum_{i=0}^{N-1} (T_i(f|_{D_i})).$$

³Ce sont habituellement des blocs images formés de carrés de différentes tailles.

Où $T_i : D_i \longrightarrow I_i$. $f|_{I_i}$ dénote la restriction de l'image f à la cellule I_i . D_i est appelée une cellule domaine et désigne un bloc appartenant à l'image mais pas nécessairement à la partition. T_i dénote une transformation de bloc élémentaire d'une cellule D_i à une cellule I_i .

Dans ce contexte, le **théorème du Collage** de la théorie des fractales (Barnsley 1988) dit que la fonction T , définie par l'ensemble des transformations élémentaires contractives $T_i, 0 \leq i < N$, pour laquelle l'image originale constitue un point fixe approximatif, possède un attracteur proche de cette image.

Par conséquent, le processus de codage consiste à trouver, pour chaque $I_i, 0 \leq i < N$, la transformation contractive T_i minimisant le critère de distorsion d .

1.2.3 Méthode de codage de base

Chaque système de codage utilise ses propres méthodes et variantes et nous décrivons brièvement à titre d'exemple celle de Jacquin (1992) dont se sont inspiré la plupart des autres méthodes. Dans ce qui suit, nous notons D un bloc de type D_i et I un bloc de type I_i .

La méthode de compression fractale (Jacquin 1992) illustrée à la figure 1.4 consiste à effectuer une recherche exhaustive sur toute l'image du meilleur bloc D à associer avec le bloc I courant.

Les blocs D subissent d'abord une transformation spatiale. Cette contrainte n'est

pas nécessaire pour la convergence de l'image décodée vers le point fixe, mais permet l'indépendance en résolution de l'image décodée (voir section 1.4). Très souvent, comme dans l'algorithme de Jacquin, les blocs D sont choisis de taille double en x et en y par rapport aux blocs I .

D'autre part, des transformations T_i de deux types sont appliquées sur les blocs D contractés spatialement et translatés:

1. *Massique*: c'est une transformation affine des intensités des pixels.
2. *Géométrique*: ce sont des isométries τ_i effectuées sur les blocs de pixels: réflexions, symétries et rotations.

Il est à noter que ces transformations n'ont aucun sens physique mais servent juste à effectuer des réarrangements de pixels variés dans le but de trouver l'approximation la plus fine possible du bloc I . La figure 1.4 illustre l'ensemble des transformations appliquées au bloc domaine D pour approximer le bloc à coder I .

Le meilleur bloc D sera celui qui minimise la distorsion (généralement l'erreur EQM) entre le bloc I_i de taille $N \times N$ et la version transformée du bloc D .

Le décodeur reçoit du canal l'ensemble des paramètres T_i ainsi que la position du meilleur bloc D_i pour reconstruire I_i . Le processus de décodage consiste à appliquer itérativement l'opérateur T déterminé lors de l'étape de codage sur une image initiale arbitraire f_0 .

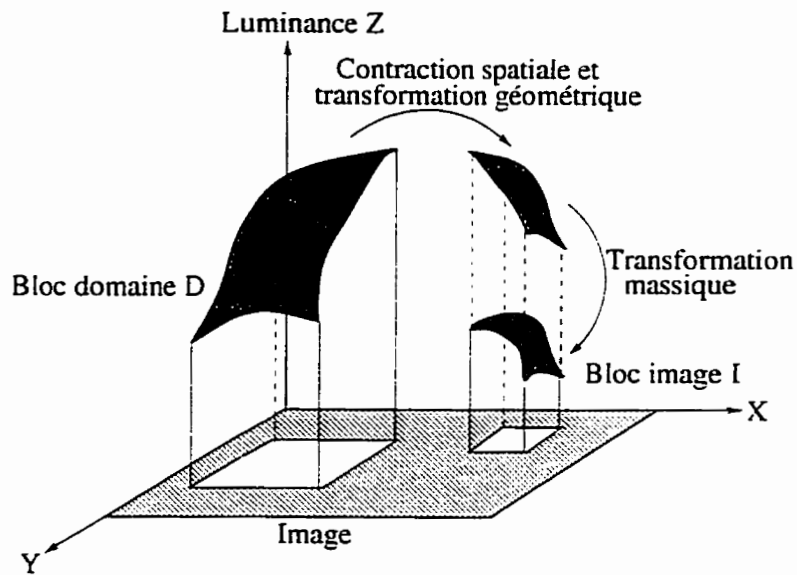


Figure 1.4 : Principe du codage fractal par bloc

1.3 Stéréoscopie et théorie de la neutralisation

1.3.1 Définitions et concepts

Vision binoculaire. Par opposition à la vision monoculaire, la vision binoculaire est un processus mettant en jeu l'usage simultané des deux yeux. Outre la perception de la profondeur, Arditi (1986) a décrit les avantages du point de vue de la sensibilité, de la reconnaissance de formes et du temps de réaction que la vision binoculaire procure par rapport à une vision monoculaire simple.

Les images stéréoscopiques correspondent aux images perçues par nos deux yeux lorsque nous observons le monde environnant. La majeure partie de la scène est donc visible des deux points de vue et se projette en des régions similaires sur les plans image gauche et droit. Ce sont donc des images très semblables dans lesquelles les

différences appelées **disparités** permettent au système visuel humain de reconstituer la dimension de profondeur et de synthétiser la sensation de relief (Arditi 1986). Le mécanisme exact de la fusion est inconnu, mais il est légitime de penser que cette synthèse se fait en appariant des régions de chaque image correspondant à la même région physique de la scène (Labonté et Laganière 1996).

Fusion binoculaire. C'est un processus cérébral par lequel deux stimulus rétiniens assez semblables sont intégrés pour aboutir à la sensation d'un objet visuel unique.

Diplopie. Quand les disparités entre les images sont trop grandes, le cerveau n'est plus capable de fusionner les deux images. L'observateur peut alors être sujet à la diplopie ou double vision, phénomène pouvant exister conjointement à la sensation de profondeur stéréoscopique.

Rivalité binoculaire et suppression. Ces phénomènes sont perceptibles lorsque deux images suffisamment différentes sont présentées simultanément aux deux yeux. Un œil prend le dessus sur l'autre et est appelé "dominant" tandis que le cerveau supprime l'information provenant de l'autre œil qui est ainsi neutralisé temporairement. Chez un individu possédant une vision normale, le mécanisme de la rivalité binoculaire entre en jeu en permanence et chaque œil devient tour à tour dominant et dominé de manière locale. Ce mécanisme serait une protection du cerveau qui générerait ainsi une section d'image unique (celle de l'œil dominant localement) dans les zones de l'image où une trop grande différence interdit l'appariement stéréoscopique et pourrait générer de la diplopie.

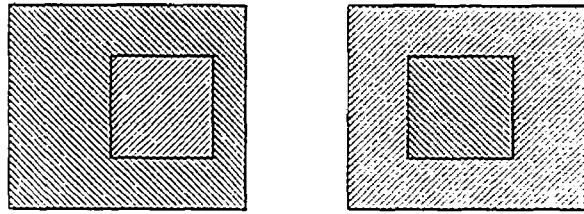


Figure 1.5 : La rivalité binoculaire ne détruit pas la stéréopsie

1.3.2 Phénomène de rivalité binoculaire

Beaucoup d'études ont été faites sur les mécanismes et les conditions dans lesquelles se produisent les phénomènes de rivalité binoculaire et de suppression (Arditi 1986, Julesz 1971). En outre, Ardit (1986) a montré un fait intéressant: *la rivalité binoculaire ne détruit pas la stéréopsie*. Comme montré à la figure 1.5, la profondeur est perceptible alors que tous les contours de ce stéréogramme sont en situation de rivalité. Ainsi, à n'importe quel instant, l'image fusionnée résultante de ce stéréogramme consiste en des régions visibles par alternance de chacune des deux images. Pourtant, la fusion met en évidence un cube 3D. Donc, malgré le fait que les deux images soient en rivalité, elles contribuent toutes les deux à l'effet stéréoscopique.

Dans la section 1.3.3, nous montrons comment ces propriétés de la vision binoculaire humaine peuvent être intégrées dans un système de codage d'images stéréoscopiques.

1.3.3 Utilisation de la théorie de la neutralisation pour le codage d'images stéréoscopiques

Deux théories s'opposent quant aux mécanismes de la vision binoculaire humaine et conduisent à des stratégies de codage différentes:

1. *la théorie de la fusion de la perception de la profondeur:*

Cette théorie (Blake 1989) suppose que les informations fournies par les deux yeux se combinent pour former un modèle mental volumétrique du monde environnant. Cette théorie amène donc à coder les deux images stéréoscopiques de manière *symétrique* impliquant la création d'une représentation intermédiaire intégrant l'information des deux images d'une paire stéréoscopique. Avec ces approches, les erreurs de reconstruction résultantes sont également distribuées entre les deux images. La représentation intermédiaire peut prendre diverses formes, comme par exemple une autre image obtenue en combinant les deux images stéréoscopiques originales ou un modèle tridimensionnel intégrant l'information des deux points de vue.

2. *la théorie de la suppression d'une des deux images en rivalité:*

Selon Dinstein, Guy, Rabany, Tselgov et Henik (1989), cette deuxième théorie conduit à l'approche suivante: "Une image de la paire stéréoscopique devrait retenir les détails de la scène tandis que la seconde image peut représenter seulement l'information de disparité. Elle peut donc être compressée sévèrement sans

« affecter la perception de la profondeur et de détail de la paire stéréoscopique. » Cette théorie inspire donc les approches de codage dites *asymétriques*, où les deux images d'une paire stéréoscopique sont traitées différemment, généralement en utilisant une image comme une référence pour la prédiction de l'autre.

1.4 Discussion

Cette revue bibliographique a permis de dégager les caractéristiques essentielles des trois approches de codage dont les méthodes sont décrites plus en détail dans l'annexe A.

L'application de la compensation du mouvement basée bloc à modèle de mouvement translationnel a permis de mettre au point des systèmes de codage vidéo en temps réel (*MPEG* et *H-261*). Cependant, le partitionnement de l'images en blocs de taille fixe et les restrictions imposées sur le type de mouvement dans la scène génèrent des artefacts dont le plus connu est l'effet de bloc. De ce fait, il devient nécessaire de transmettre l'erreur de reconstruction pour pouvoir garder une qualité acceptable des images perçues au récepteur. Ceci diminue le pouvoir de compression d'autant plus que cette erreur n'est, en général, pas très corrélée spatialement et n'a, par conséquent, pas beaucoup de potentiel pour le codage.

Pour pallier à ces inconvénients, des stratégies par blocs de taille variable, par régions polygonales, par régions de forme arbitraire à modèle de mouvement ou par compensation du mouvement dense ont été proposées. Ces stratégies permettent d'améliorer

la qualité des images reconstruites mais au détriment de la compression et surtout de la simplicité du codeur.

En ce qui concerne le codage fractal, ses qualités majeures résident dans la simplicité et la rapidité de son système de décodage et dans un remarquable pouvoir de compression, en particulier pour des applications à très bas débit binaire. En outre, une propriété exclusive du codage fractal est l'indépendance en résolution de l'image décodée⁴. D'autre part, la modélisation fractale, basée sur une stratégie itérative convergente ne requiert pas la transmission de l'erreur de prédiction entre l'image originale et l'image reconstruite. Egalement, le type de transformations considérée rend l'approche par codage fractal indépendante de la complexité du mouvement dans la scène contrairement aux méthodes par compensation du mouvement basées blocs. Par contre, une première faiblesse de l'application pratique de la théorie des fractales provient de la limitation imposée pour des raisons de complexité d'implantation sur les types de transformations considérés, la taille des blocs codés et le rapport de contractivité spatiale. Ceci fige considérablement le pouvoir de détection des redondances fractales dans l'image (Ratnakar, Feig et Tiwari 1994). Finalement, le désavantage majeur des stratégies de codage fractal réside dans le fait qu'elles sont dans leur très grande majorité⁵ basées sur une mise en correspondance. Par conséquent, elles sont très coûteuses en temps de calcul.

⁴Le codage fractal permet de générer des détails artificiels à toutes les échelles dans une image.

⁵Les autres méthodes concernent des applications à très bas débit binaire.

Enfin, les stratégies de codage asymétrique de séquences d'images stéréoscopiques intégrant la théorie de la neutralisation n'ont pas encore été suffisamment développées en dépit de leur grand potentiel. En effet, le fait qu'un œil puisse recevoir une image à basse résolution sans que cela pénalise la perception globale est familier à de nombreuses personnes. Il a été montré (Zhu, Gao et Goutte 1991, Perkins 1992) que les personnes possédant un œil bon et l'autre faible⁶ perçoivent le monde avec la qualité du meilleur œil.

Dans ce mémoire, nous proposons un système de compression asymétrique de séquences d'images stéréoscopiques combinant la compensation du mouvement et le codage fractal. Une méthodologie hybride va permettre de conjuguer les forces des deux approches de codage. Dans le chapitre 2, nous présentons la structure générale de notre système et la justification des choix de notre stratégie de codage.

⁶sans être cependant amblyopes, auquel cas la vision stéréoscopique disparaît.

Chapitre 2

Présentation générale du système

Dans ce chapitre, nous présentons les caractéristiques que doit satisfaire notre système. Puis, à partir de ces contraintes, nous exposons les réflexions ainsi que les justifications qui nous ont menés vers le choix de notre méthode de codage de séquences d'images stéréoscopiques.

Nous avons choisi de développer un système de codage asymétrique de séquences d'images stéréoscopiques. Aussi, par la suite, nous notons H la séquence d'images codée à haute résolution et L la séquence d'images homologue de H codée à faible résolution.

La structure du codeur que nous avons développé est présentée à la figure 2.1. Chaque image de la séquence (de type H ou L) présentée à l'entrée du codeur est d'abord partitionnée en blocs non superposés. Puis, ces blocs passent successivement—si besoin est—dans des modules d'estimation des blocs actifs, de codage par compensation

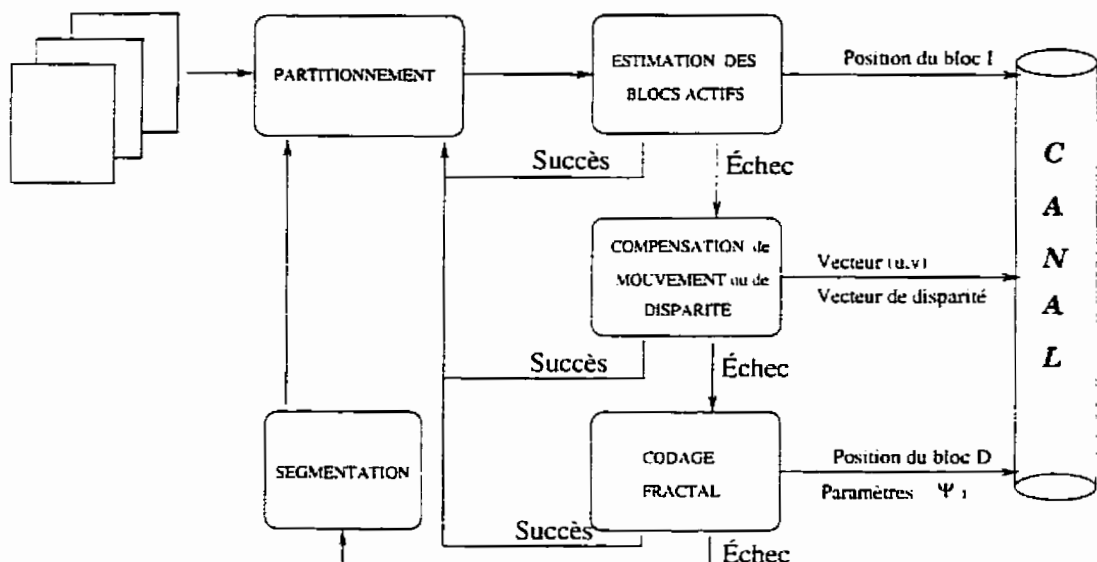


Figure 2.1 : Schéma bloc du codeur

du mouvement ou de disparité (possible pour la séquence L seulement), de codage fractal et de segmentation en arbre quaternaire.

La justification du choix et de l'ordre des différents éléments du système figurent dans les sections 2.2 et 2.3. Auparavant, nous présentons les critères que doit remplir notre système.

2.1 Description des critères

1. Nous cherchons à développer un système de compression de séquences d'images stéréoscopiques tel que la qualité footnotéevaluée selon des critères objectifs et subjectifs, d'une des deux séquences (celle à haute résolution H) soit comparable, pour des taux de compression moyens identiques, à celle résultant d'un codage de type *MPEG*.

2. Le taux de compression obtenu pour le codage des deux séquences doit se rapprocher de celui obtenu pour coder une séquence monoculaire. Ainsi, en considérant qu'une des deux séquences doit respecter le premier critère, la deuxième séquence (celle à faible résolution L) doit pouvoir être compressée très sévèrement tout en assurant une qualité de fusion footnoteévaluée selon des critères perceptuels. satisfaisante.
3. Les critères de simplicité algorithmique et de rapidité d'exécution du codage doivent être également pris en compte dans un but futur d'implantation matérielle en temps réel.

2.2 Codage de la séquence à haute résolution H

Pour satisfaire le premier critère, il est judicieux d'exploiter au maximum tous les types de répétition d'information disponibles dans une séquence d'images. Comme expliqué dans l'introduction, ceux-ci se présentent sous la forme de redondance spatiale, temporelle et fractale. La satisfaction du dernier critère nous oriente vers une méthode de compression vidéo basée blocs bi-dimensionnels. En effet, dans le chapitre 1, nous avons vu que toutes les autres méthodes plus originales de compression vidéo utilisant un codage par compensation du mouvement ou un codage fractal basées régions ou utilisant des blocs tri-dimensionnels (Diab et Cohen 1996, Frobert et Cohen 1996a) ont un coût de calcul trop élevé pour que nous puissions les considérer.

A ce niveau, comme nous voulons exploiter la redondance temporelle dans la séquence, la première question qui se pose est :

Comment capturer le mouvement dans la séquence vidéo ?

Nous avons vu que les normes de compression vidéo en temps réel se basaient sur des techniques de compensation du mouvement par blocs à modèle de mouvement translationnel (voir section 3.5.2.1) avec transmission de l'erreur de prédiction. Cependant, la qualité visuelle des images reconstruites est entachée du phénomène d'effet de bloc décrit dans la section 1.1. De plus, l'accumulation d'erreur inhérente aux méthodes de compensation du mouvement obligent la transmission à intervalles relativement proches d'images de mise à jour: ceci occasionnant une diminution de la compression globale.

Pour satisfaire les objectifs de qualité et de compression, nous cherchons à développer un système de codage qui, en plus d'être moins source d'effet de bloc et d'accumulations d'erreurs que les systèmes existant, présenterait des résultats équivalents en qualité et compression pour toute séquence d'images *quel que soit le type de mouvement dans la scène*.

Cette caractéristique est primordiale car les méthodes basées sur la compensation du mouvement seul sont sensibles à la complexité des mouvements dans la scène. En effet, ces méthodes utilisant le modèle translationnel génèrent une erreur de prédiction importante coûteuse à transmettre.

lorsque la scène est animée d'un mouvement plus complexe (rotation, divergence etc.) par rapport à la caméra.

Cette considération nous amène à utiliser une modélisation fractale du mouvement car, comme nous l'avons déjà dit dans le chapitre 1, les transformations issus du codage fractal ne représentent pas un modèle de mouvement physiquement existant. En fait, la modélisation fractale exploite le contenu de l'image pour trouver la meilleure approximation possible d'un bloc, mais de manière indépendante du mouvement de la scène.

En outre, les méthodes par codage fractal sont suffisamment élaborées pour approximer des blocs complexes et ne requièrent pas nécessairement le codage de l'erreur de prédiction sous réserve que l'approximation soit suffisamment fine (Fisher, Shen et Rogovin 1994*b*). Ceci offre un plus grand potentiel de compression par rapport aux méthodes de codage par compensation du mouvement.

Cependant, nous avons vu qu'il existait des méthodes très variées de codage d'images et de séquences vidéos utilisant la théorie des fractales seule ou combinée avec d'autres méthodes de compression existantes. La deuxième question qui se pose est:

Quel type de codage fractal utiliser ?

Nous devons remplir le dernier critère de simplicité et de rapidité. Nous avons donc choisi d'effectuer un codage fractal inter-frames pour assurer un fonctionnement causal du système. Ceci signifie que les blocs domaines

utilisés pour coder les blocs images appartiennent à l'image précédente et sont donc disponibles au décodeur. De ce fait, le décodage est instantané et non itératif et la qualité de l'image reconstruite au décodeur peut être calculée directement à l'émetteur. Ceci nous permet donc d'adapter certains seuils lorsque la qualité ou le taux de compression obtenus ne sont pas satisfaisants. De plus, ceci nous libère de la contrainte de contractivité.

Cependant, le codage fractal demeure une étape très coûteuse en temps de calcul au codeur. Nous avons cherché le meilleur moyen de réduire ce temps de calcul sans pénaliser la qualité de l'image reconstruite. Ainsi, il nous a semblé pertinent de ne pas effectuer un codage fractal pur, mais de le cumuler avec d'autres méthodes de codage.

Les études (Andonova et Popovic 1994) et (de Faria et Ghanbari 1995) ont prouvé qu'un codage hybride compensation du mouvement codage fractal donnait de piètres résultats lorsque le codage fractal était utilisé pour coder l'erreur de reconstruction généré par la compensation du mouvement. Par contre, lorsqu'un module de compensation du mouvement est d'abord effectué sur les blocs et que les blocs non estimables sont codés de manière intra-trame par codage fractal, les résultats obtenus sont très encourageants.

Nous avons donc choisi d'utiliser une méthode quasi-similaire à la dif-

férence près que, comme Fisher et al. (1994b), le codage fractal est effectué de manière inter-frames.

Notre appellation de codage "hybride" réfère donc à l'utilisation conjointe de la compensation du mouvement et du codage fractal et non à la combinaison de codage intra et inter-frames.

En fait, l'avantage de combiner les deux méthodes de codage est d'améliorer la qualité résultante pour un taux de compression donné par rapport à une méthode de compensation du mouvement seul ou de codage fractal seul. Le chapitre 4 analysera qualitativement et quantitativement l'amélioration obtenue.

Par ailleurs, pour accélérer l'étape de codage fractal, nous utilisons un domaine de recherche limité des blocs domaines et nous travaillons dans l'espace des fréquences (Zhao et Yuan 1994). En effet, la transposition de la recherche du meilleur bloc domaine dans l'espace fréquentiel nous permet de réduire le temps de calcul par la construction d'une représentation invariante aux transformations fractales (Wohlberg et de Jager 1995). Cette technique permet de réduire de façon substantielle le temps de codage.

En outre, préalablement au module de compensation du mouvement, une recherche des blocs actifs est menée. Nous montrerons dans le chapitre suivant qu'elle permet d'augmenter de manière non négligeable le taux de compression tout en réduisant le temps de calcul car le codage n'est mené que sur les blocs actifs. Enfin, nous avons

choisi de mener un système de codage multi-niveaux avec décomposition en arbre quaternaire pour le gain de compression supplémentaire qu'il permet d'obtenir.

L'ordonnancement des différents modules est choisi de manière à effectuer prioritairement les modules les plus rapides et générant la compression la plus importante et à obtenir une approximation de plus en plus fine au fil des modules. En effet, le module d'estimation des blocs actifs constitue une tentative de compensation du mouvement à modèle nul et génère donc le maximum de compression pour une qualité plus grossière. Ensuite, le module de compensation du mouvement à modèle translationnel raffine l'approximation et, s'il s'avère inefficace, est suivi d'un module de codage à modélisation fractale plus précis, mais plus long et nécessitant la transmission de plus d'information. Enfin, le module de segmentation divise un bloc trop complexe à coder en 4 sous-blocs, diminuant ainsi la compression et augmentant le temps de calcul pour améliorer la qualité.

2.3 Codage de la séquence à faible résolution L

La satisfaction des deux premiers critères justifie *a posteriori* le choix d'une méthode de codage asymétrique des deux séquences homologues. En effet, le premier critère oblige à conserver une des deux séquences avec une qualité suffisamment bonne pour être visualisée monoculairement. Ceci interdit les méthodes de codage symétriques dans laquelle les deux séquences sont dégradées de façon significative (Labonté et Laganière 1996). Le deuxième critère du cahier des charges

spécifie que la séquence stéréoscopique compressée ne doit pas représenter plus d'information qu'une séquence monoculaire compressée. La compression globale de l'information doit donc être comparable à la transmission d'une seule séquence. Nous devons donc utiliser un système de codage asymétrique utilisant la théorie de la neutralisation. Comme expliqué dans la section 1.3.3, selon cette théorie, la compression sévère mais contrôlée d'une des deux images d'une paire stéréoscopique place l'observateur en situation de rivalité binoculaire, mais sans interrompre la fusion globale des deux images. L'application de cette théorie au codage de séquences d'images stéréoscopiques consiste donc à coder les deux séquences à des résolutions très différentes. Comme méthode de codage, suite aux expériences décrites dans la section A.3.2 et pour satisfaire au critère de taux de compression minimum de L_t , nous avons choisi de coder une version filtrée passe-bas de la séquence à basse résolution. Ce filtrage sous-échantillonnage est réalisé au moyen d'une pyramide gaussienne (voir section A.3.1) et permet de réduire la visibilité des distorsions dues à la compression plus importante de la séquence à faible résolution.

Ensuite, le même type de codage hiérarchique que pour la séquence à haute résolution est effectué sauf qu'un module de compensation de disparité (voir A.3.1) peut remplacer la compensation du mouvement. Ceci est justifié par le fait que, en supposant les caméras stéréoscopiques alignées, la recherche du meilleur vecteur de disparité se limite à une recherche uni-dimensionnelle c'est-à-dire à un déplacement horizontal contrairement à la compensation du mouvement. De ce fait, le taux de compression

obtenu est plus important et le temps de calcul diminué.

Il est à noter que le module de compensation de disparité n'est pas indispensable à la méthode. En effet, nous tenons déjà compte implicitement de la nature stéréoscopique des séquences grâce au principe de neutralisation. La seule raison pour laquelle une séquence peut être à basse résolution provient du fait qu'elle représente un des deux points de vue d'une paire stéréoscopique. Par conséquent, une variante de notre méthode consiste à faire un codage hybride de la séquence H à haute résolution et de faire le *même* codage hybride de la séquence L à basse résolution. Ceci présente l'avantage de pouvoir mener le codage et le décodage des deux séquences en parallèle. Une comparaison des résultats des deux méthodes est décrite dans la section 4.2.

Chapitre 3

Description de la méthode de codage

Dans ce chapitre, nous décrivons les éléments constitutifs du codeur et du décodeur tout en justifiant le choix entre certaines alternatives et les valeurs de différents paramètres utilisés dans les différents modules. Pour cela, nous présentons toute une série de courbes représentant l'évolution du rapport entre le taux de compression et l'erreur quadratique moyenne ($\frac{\tau}{EQM}$) en fonction des valeurs de ces paramètres. Les séquences d'images monoculaires et stéréoscopiques utilisées pour ces expérimentations sont présentées dans l'annexe B. L'annexe D comprend des courbes supplémentaires pour interpréter les effets de ces paramètres sur le taux de compression et sur l'erreur quadratique moyenne obtenus séparément.

Notre système, tel que montré à la figure 3.1 comporte deux parties distinctes: celle

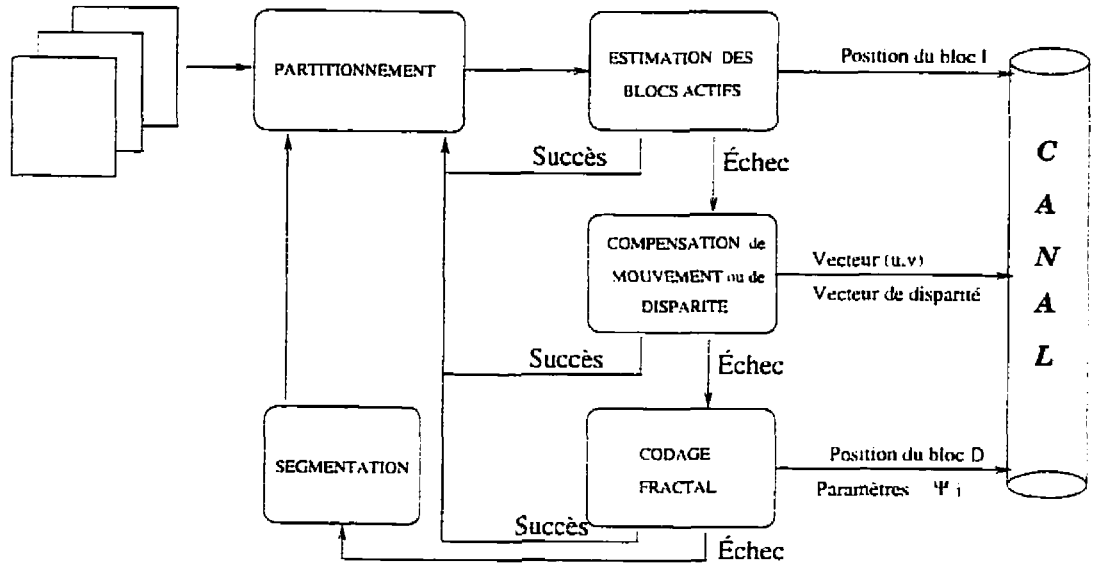


Figure 3.1 : Schéma bloc du codeur

relative au codage de l'image à haute résolution H_t et celle relative à une variante possible du codage de l'image homologue à basse résolution L_t . La première partie, décrite dans la section 3.1, pourrait être utilisée indépendamment pour coder des séquences d'images monoculaires. La section 3.2 décrit les particularités supplémentaires caractéristiques du codage de la séquence L_t . Enfin, la section 3.5 présente les caractéristiques principales de l'implantation sur ordinateur de notre système ainsi que le codage de source requis pour le rendre plus résilient aux erreurs de canal.

3.1 Codage de la séquence à haute résolution H

Chaque image H_t est partitionnée en blocs non superposés et codée de manière inter-frames à partir de l'image précédente reconstruite \hat{H}_{t-1} . Cette façon de faire permet de diminuer les erreurs de reconstruction en utilisant une structure symétrique

au codeur et au décodeur.

Dans la suite, nous notons $B_t(x, y)$ la valeur du pixel du bloc B_t de taille $N \times N$ et de position (x, y) dans l'image au temps t et \hat{B}_t un bloc de l'image reconstruite au temps t .

3.1.1 Module d'estimation des blocs actifs

Chaque bloc de l'image est comparé au bloc de la même position de l'image précédente. Cette comparaison permet d'évaluer si le contenu du bloc a beaucoup varié ou non entre les deux images consécutives. Seuls les blocs classés actifs (correspondant à des régions en mouvement de la scène) seront codés par les modules suivants. Un bloc inactif (inchangé entre les deux images consécutives) sera reconstruit au décodeur en utilisant le bloc de la même position de l'image précédente.

À ce niveau, une question essentielle se pose :

Quel critère d'activité utiliser ?

En effet, il faut s'assurer que :

- tous les blocs correspondant à des zones en mouvement de la scène soient classés comme actifs;
- trouver un critère insensible au bruit présent dans l'image.

La seconde considération nous amène à calculer la gamme d'activité de blocs existant sur toute l'image et de fixer un seuil d'activité dépendant

de cette gamme.

Ainsi, pour évaluer l'activité du bloc B_t de taille $N \times N$ de position (x, y) , le paramètre suivant est déterminé :

$$var_N(B_t) = \sqrt{\frac{1}{N^2} \sum_{(k,l)=(0,0)}^{(N-1,N-1)} \left[B_t(x+k, y+l) - \hat{B}_{t-1}(x+k, y+l) \right]^2}. \quad (3.1)$$

Ce paramètre est calculé dynamiquement sur tous les blocs de l'image et pour chaque taille N de bloc. Ensuite, les variances minimale var_N^{min} et maximale var_N^{max} de l'ensemble des blocs B_t de taille $N \times N$ de l'image sont déduites.

Nous pouvons alors calculer le seuil $\Gamma_{act,N}$ utile selon l'expression :

$$\Gamma_{act,N} = var_N^{min} + (var_N^{max} - var_N^{min})x_t. \quad (3.2)$$

où x_t représente un pourcentage de la répartition des blocs actifs considérés dans l'image au temps t . La valeur de ce seuil évolue avec le temps.

Un bloc B_t de taille $N \times N$ est finalement considéré comme actif si $var_N(B_t)$ est supérieure à $\Gamma_{act,N}$.

Étude de l'influence du seuil x_t : Lorsque x_t augmente, $\Gamma_{act,N}$ augmente et donc le nombre de blocs actifs à coder diminue. Le taux de compression est plus élevé, mais au détriment de la qualité de l'image reconstruite.

Lorsque x_t diminue, l'inverse se produit.

Pour chaque image t de chaque séquence, il existe une valeur optimale de x_t donnant le meilleur compromis qualité/taux de compression, mais celle-ci n'est pas connue a priori.

Par conséquent, nous avons choisi d'initialiser x_0 à une valeur assez petite (10) en début de séquence et de l'adapter de façon dynamique selon le taux de compression (noté τ) et deux erreurs de reconstruction obtenues à la trame précédente. Nous considérons l'erreur quadratique moyenne sur l'image globale reconstruite et sur les blocs actifs reconstruits (notés EQM_t et $EQM_{act,t}$).

$$EQM_t = \frac{1}{Nb} \sum_{Nb} \sqrt{\frac{1}{N^2} \sum_{(k,l)=(0,0)}^{(N-1,N-1)} \left[B_t(x+k, y+l) - \hat{B}_{t-1}(x+k, y+l) \right]^2} \quad (3.3)$$

$$EQM_{act,t} = \frac{1}{Na} \sum_{Na} \sqrt{\frac{1}{N^2} \sum_{(k,l)=(0,0)}^{(N-1,N-1)} \left[B_t(x+k, y+l) - \hat{B}_{t-1}(x+k, y+l) \right]^2} \quad (3.4)$$

où Nb et Na représentent respectivement le nombre de blocs total et le nombre de blocs actifs B_t de l'image. L'adaptation se fait en utilisant les formules suivantes :

$$x_t = \begin{cases} x_{t-1} + \delta x & \text{si } \tau \leq 50. \\ x_{t-1} - \delta x & \text{si } EQM_t \geq 5 \text{ et } EQM_{act,t} \geq 10. \\ x_t & \text{sinon.} \end{cases} \quad (3.5)$$

avec $\delta x = 1$.

Il faut prendre en compte l'erreur $EQM_{act,t}$ car, si cette dernière est trop élevée,

elle peut générer des régions de l'image de qualité visuelle très mauvaise et gênante. En outre, si l'erreur $EQM_{act,t}$ est assez faible, il faut quand même prendre en compte l'erreur EQM_t globale car elle contient l'erreur issue des blocs inactifs. Ainsi, nous contrôlons la limitation d'accumulation d'erreur sur les blocs inactifs.

Les valeurs choisies comme condition d'adaptation de x_t sont arbitraires et, dans notre cas, ont été choisies de façon empirique.

Le module d'estimation des blocs actifs permet donc d'augmenter grandement le taux de compression et la vitesse du codeur en diminuant le nombre de blocs à considérer. De plus, un ajustement adaptatif du pourcentage x permet d'effectuer un contrôle du débit binaire et de l'erreur.

3.1.2 Module de compensation du mouvement

3.1.2.1 Principe

Pour des raisons de simplicité du codeur, nous avons choisi d'utiliser une méthode de compensation du mouvement à modèle translationnel. Comme décrit dans la section 1.1.3, l'estimation de mouvement associe pour chaque bloc B_t de l'image courante à coder un vecteur représentant le déplacement de ce bloc entre les deux images successives. Ce vecteur correspond à une translation parallèle à la caméra ce qui revient donc à rechercher pour un bloc donné le meilleur bloc \hat{B}_{t-1} de même taille translaté à une autre position dans l'image précédente reconstruite.

Nous avons choisi de travailler dans un espace de recherche adaptatif en spirale de

longueur λ_{mvt} débutant au pixel du coin supérieur gauche du bloc courant à coder, mais dans l'image précédente.

Le pas de la spirale est de un pixel. De plus, nous nous contentons d'une précision au niveau pixel pour éviter d'avoir à faire une interpolation au décodeur. La figure 3.2 illustre le principe générique de notre module de compensation du mouvement.

Pour chaque bloc $B_t(x, y)$ de taille $N \times N$, le meilleur bloc $\hat{B}_{t-1}(x + u, y + v)$ appartenant à la spirale λ_{mvt} satisfait le critère suivant :

$$EQM = \min_{\lambda_{mvt}} \sqrt{\sum_{(k,l)=(0,0)}^{(N-1,N-1)} \frac{1}{N^2} \left[B_t(x + k, y + l) - \hat{B}_{t-1}(x + u + k, y + v + l) \right]^2}. \quad (3.6)$$

Si cette erreur est inférieure ou égale à un seuil donné Γ_{mvt} (dont le choix est discuté dans la section 3.1.2.2), la compensation du mouvement est satisfaisante et le vecteur de déplacement (u, v) est transmis au canal. Sinon, le bloc B_t passe dans le module de codage suivant.

3.1.2.2 Étude de l'influence des seuils et paramètres

Cette section compare les résultats obtenus en faisant varier les valeurs internes et le niveau de quantification des paramètres utilisés par ce module tout en gardant fixes les autres paramètres des différents modules. Il est à noter que les autres modules ne sont pas désactivés car ils interviennent dans la performance globale du système. En fait, la difficulté de cette étude provient de l'interaction mutuelle des paramètres.

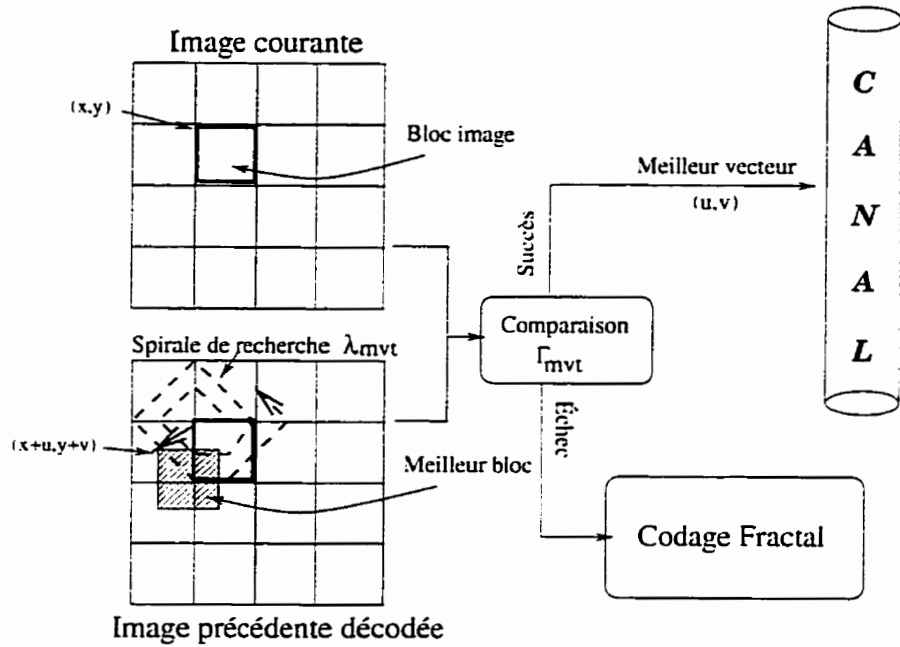


Figure 3.2 : Module de compensation du mouvement

1. *Influence du seuil de succès du module Γ_{mvt} :*

Pour toutes les séquences de la figure 3.3, nous pouvons observer que le meilleur rapport $\frac{\tau}{EQM}$ est obtenu pour $\Gamma_{mvt} = 8$.

$\Gamma_{mvt} = 4$ est clairement un mauvais choix car, comme nous pouvons le constater sur les figures D.1 et D.2, il diminue trop la compression pour une amélioration de l'erreur quadratique moyenne globale trop faible.

Pourtant, nous avons très souvent fixé $\Gamma_{mvt} = 6$. En effet, il faut voir que prendre $\Gamma_{mvt} = 8$ peut résulter en des artefacts visuels gênants car le module de compensation du mouvement réussit à coder des plus gros blocs. Les deux valeurs sont donc à considérer selon l'application.

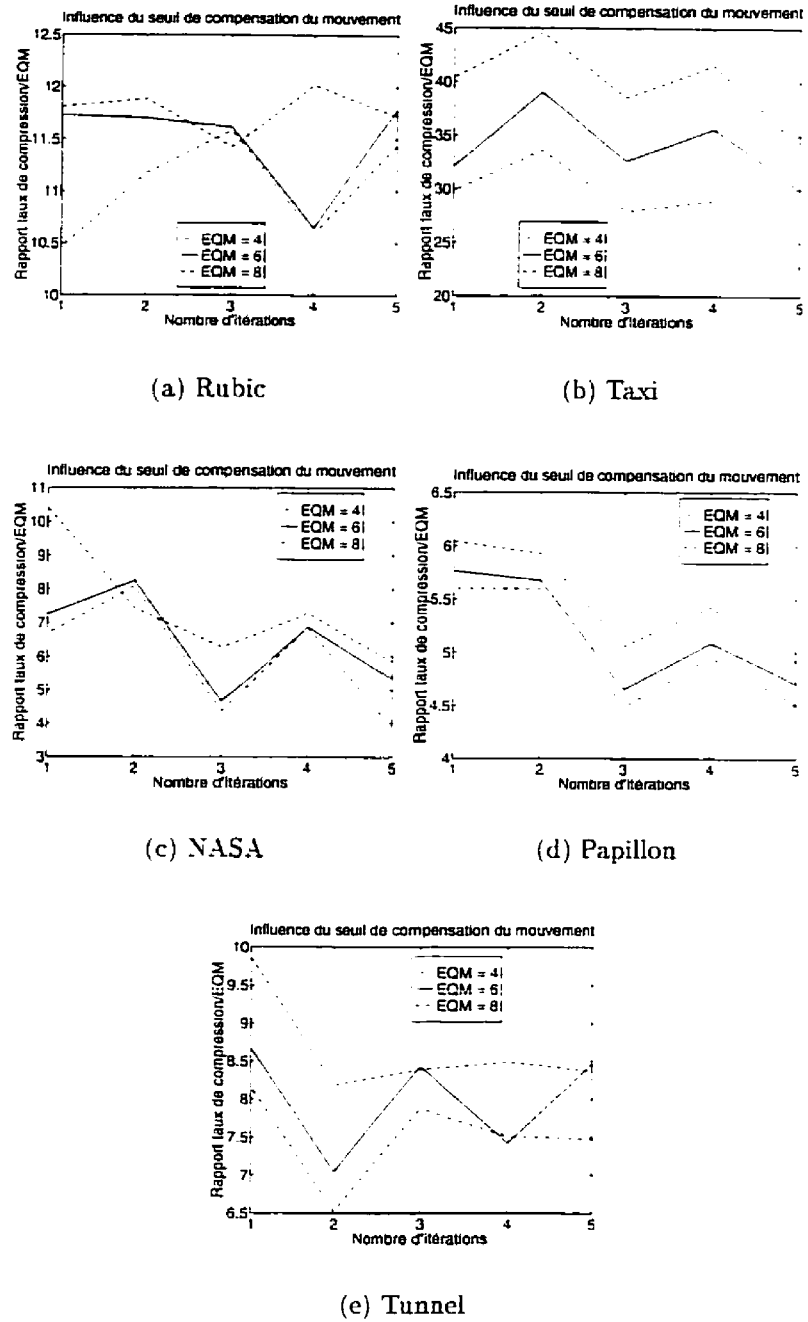


Figure 3.3 : Influence du seuil de succès du module de compensation du mouvement Γ_{mvt} sur le rapport entre le taux de compression et l'erreur quadratique moyenne

2. Influence de la longueur de la spirale de recherche λ_{mvt} :

Les résultats illustrés à la figure 3.4 (où λ_{mvt} est donné en nombre de bits) diffèrent selon les séquences. Pour la séquence "NASA", aucune courbe ne se détache des autres. Par contre, les résultats obtenus avec les séquences "Taxi" ."Papillon"."Tunnel" et "Rubic" suggèrent de prendre $\lambda_{mvt}=32$ ou 64. En effet, considérer $\lambda_{mvt} = 16$ amène à compenser la faiblesse du module de compensation du mouvement par le codage fractal d'un plus grand nombre de blocs. Ceci résulte, comme nous pouvons le voir sur les figures D.3 et D.4, en une baisse du taux de compression et une hausse de l'erreur quadratique moyenne.

Au contraire, prendre $\lambda_{mvt}=128$ donne l'erreur quadratique moyenne la plus faible, mais peut diminuer la compression inutilement car une telle valeur ne représente plus un déplacement plausible entre deux images consécutives (car il serait trop important), mais plutôt un codage fractal simplifié de rapport de contractivité 1.

3.1.3 Module de codage fractal

Le module de codage fractal, cœur de notre méthode, est utilisé pour tous les blocs classés actifs pour lesquels la tentative de compensation du mouvement a échoué.

La figure 3.5 illustre le principe simplifié de notre module de codage fractal. Il va consister principalement en trois étapes: la construction d'une représentation inva-

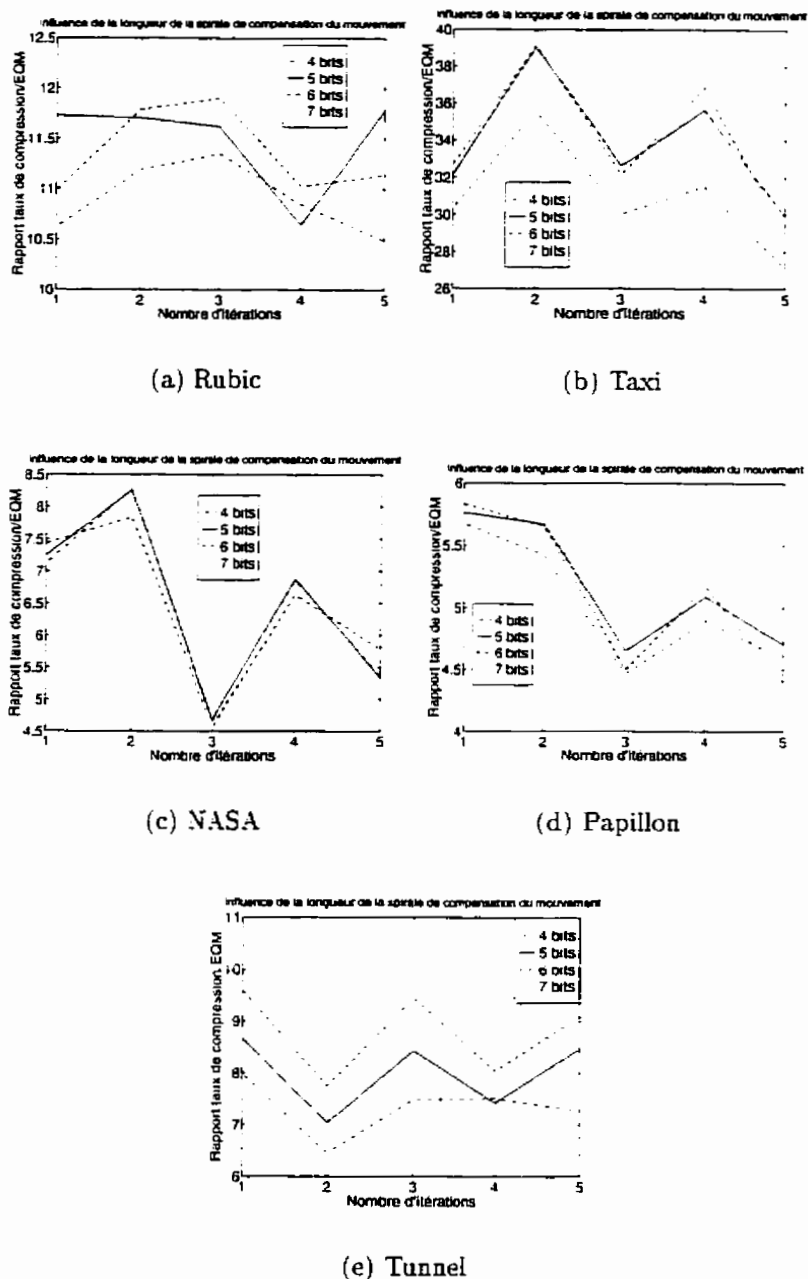


Figure 3.4 : Influence de la longueur de la spirale de recherche λ_{mvt} sur le rapport entre le taux de compression et l'erreur quadratique moyenne

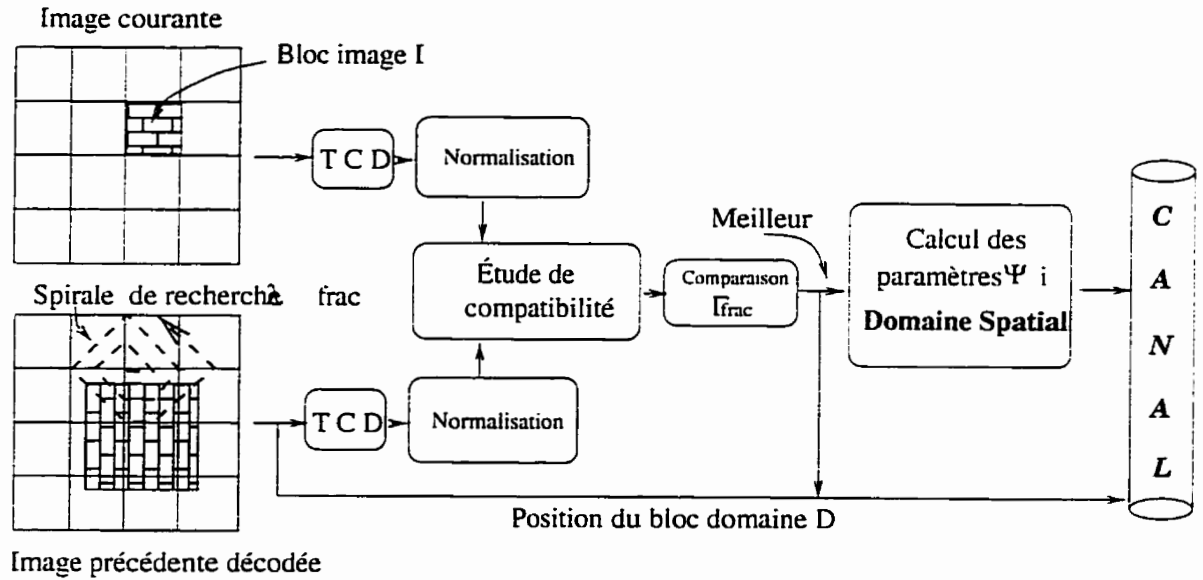


Figure 3.5 : Module de codage fractal

riante des blocs de l'image, la recherche du meilleur bloc approximant le bloc à coder et la détermination des paramètres de la transformation fractale associée.

3.1.3.1 Domaine de recherche des blocs D

Comme nous effectuons un codage inter-frames, le domaine de recherche des blocs D pouvant approximer un bloc image I est situé dans l'image précédente par rapport à l'image à laquelle appartient le bloc I à coder. Au lieu de considérer une recherche exhaustive trop coûteuse en temps de calcul, nous choisissons un espace de recherche limité en spirale de longueur λ_{frac} centrée autour de la position du bloc I courant, mais dans l'image précédente. Le pas de la spirale dépend du niveau de segmentation. Plus les blocs considérés sont petits (limite de 4×4), plus le pas $h(N)$ entre deux blocs D consécutifs de taille $2N \times 2N$ est petit (limite de un pixel). Nous avons fait

ce choix pour privilégier la qualité au niveau de résolution le plus fin sans utiliser trop de mémoire ni trop de temps de calcul aux niveaux plus grossiers.

3.1.3.2 Espace de traitement

Nous avons choisi de mener la recherche du meilleur bloc D dans l'espace transformé correspondant à la transformée par cosinus discret ($TC D$) selon l'équation 3.7.

$$\mathcal{B}(u, v) = \frac{c(u)c(v)}{4} \sum_{(x,y)=(0,0)}^{(N-1,N-1)} B(x, y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right). \quad (3.7)$$

avec $u, v = 0, \dots, N-1$, $c(0) = 1$ et $c(u) = \frac{1}{\sqrt{2}}$ si $u \neq 0$.

Le coefficient (u, v) d'un bloc B de taille $N \times N$ transformé par $TC D$ est noté :

$$TC D_B(u, v) = \mathcal{B}(u, v), \text{ avec } u, v = 0, \dots, N-1.$$

Chaque bloc de $N \times N$ pixels $B(x, y)$ est transformé via l'opérateur $TC D$ en un bloc de $N \times N$ coefficients $\mathcal{B}(u, v)$ représentatifs du contenu fréquentiel du bloc original.

Chaque bloc image I ou bloc domaine D est donc transformé via la $TC D$ en \mathcal{I} ou \mathcal{D} respectivement. Cependant, les paramètres de codage fractal d'un bloc envoyés dans le canal sont indépendants du domaine de traitement choisi (fréquentiel ou spatial).

De fait, au décodeur, la reconstruction de chaque bloc est menée dans le domaine spatial sans avoir besoin de calculer de $TC D$ inverse. Cette méthode ne ralentit donc

pas le décodeur.

Cette nouvelle représentation du bloc dans un autre domaine présente, en outre plusieurs avantages :

1. La *TC D* permet d'exploiter la redondance spatiale existant entre les pixels d'un même bloc.
2. Les blocs transformés par *TC D* possèdent des propriétés intéressantes du point de vue invariance lorsque certaines transformations massiques et géométriques leur sont appliquées.

3.1.3.3 Transformations effectuées sur les blocs D

Pour chaque bloc I de taille $N \times N$, les blocs D candidats sont de taille fixe $2N \times 2N$.

Dans le domaine spatial, le meilleur bloc D appartenant à la spirale de longueur λ_{rac} satisfait le critère :

$$E = \min_{\Psi, \lambda_{rac}} \sqrt{\sum_{(x,y)=(0,0)}^{(N-1,N-1)} \frac{1}{N^2} [I(x,y) - \Psi \circ \phi(D)(x,y)]^2}. \quad (3.8)$$

ϕ est une fonction contractive, ce qui correspond à effectuer une contraction spatiale d'ordre 2 selon chaque axe sur le bloc D . Le bloc résultant $\phi(D)$ est de taille

$N \times N$. Nous avons choisi pour ϕ une fonction de moyennage :

$$\phi(D)(x, y) = \frac{D(x, y) + D(x+1, y) + D(x, y+1) + D(x+1, y+1)}{4}. \quad (3.9)$$

Ψ est composée de 2 types de transformations appliquées sur le bloc $\phi(D)$.

Dans un souci de simplicité, nous confondons par la suite $\phi(D)$ avec D car tous les calculs seront faits avec la version contractée de D . Ainsi, nous considérons \mathcal{D} comme étant la *TCD* de la version contractée spatialement de D .

En transformant les blocs dans l'espace *TCD*, le meilleur bloc \mathcal{D} satisfait le critère équivalent 2 :

$$E = \min_{\Psi, \lambda_{frac}} \sqrt{\sum_{(u,v)=(0,0)}^{(N-1,N-1)} \frac{1}{N^2} [I(u, v) - \Psi \circ \mathcal{D}(u, v)]^2}. \quad (3.10)$$

En effet, cette relation est équivalente à la précédente (relation 3.8) car la transformation Ψ considérée est composée des transformées affines et isométriques. Elle conserve donc les distances aussi bien dans le domaine spatial que fréquentiel.

1. Expression des transformations Ψ dans le domaine spatial :

- (a) *transformation massique* de type affine sur les intensités des pixels. Cette transformation s'exprime par :

$$\begin{aligned} \mathcal{R} &\longrightarrow \mathcal{R} \\ z(x, y) &\longmapsto az(x, y) + b, \end{aligned} \quad (3.11)$$

où $z(x, y)$ désigne la luminance du pixel (x, y) : a correspond à un changement d'échelle et b à un décalage.

- (b) *transformation géométrique*: les 8 isométries numérotées de 0 à 7 sont représentées dans la colonne gauche de la figure 3.6.

2. Transposition dans l'espace des fréquences :

- (a) Dans le domaine TCD , le coefficient (u, v) d'un bloc B de taille $N \times N$ sur lequel est effectué une transformation massique a pour expression :

$$TCD_{aB+b}(u, v) = a\mathcal{B}(u, v) + bN\delta_0^u\delta_0^v. \quad (3.12)$$

où δ_i^j désigne le symbole de Kronecker.

- (b) Dans la figure 3.6, nous notons \mathcal{T} le résultat de l'application d'une isométrie massique ou géométrique sur le bloc \mathcal{B} . Cette figure montre que, dans l'espace des fréquences, les transformations géométriques entraînent des modifications mineures de la TCD du bloc original avant transformation. Ainsi, par exemple, l'isométrie 5 consistant à effectuer une rotation de -90 degrés du bloc B original s'exprime dans l'espace des fréquences par deux modifications de \mathcal{B} . En notant toujours $\mathcal{B}(u, v)$ le coefficient (u, v) du bloc \mathcal{B} , la modification due à l'application de l'isométrie 5 sur le bloc \mathcal{B} revient à permuter les rôles de u et v et à inverser les signes des valeurs des coefficients pour u impair.

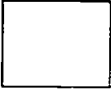

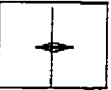
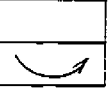
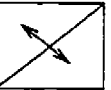

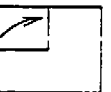
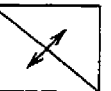
	DOMAINE SPATIAL	DOMAINE TCD
Transformation Géométrique	0  Identité	$T_{(u,v)} = B_{(u,v)}$
	1  Réflexion / demi-axe vertical	$T_{(u,v)} = (-1)^v B_{(u,v)}$
	2  Réflexion / demi-axe horizontal	$T_{(u,v)} = (-1)^u B_{(u,v)}$
	3  Rotation 180 degrés	$T_{(u,v)} = (-1)^{u+v} B_{(u,v)}$
	4  Réflexion / 2 ^e diagonale	$T_{(u,v)} = (-1)^{u+v} B_{(v,u)}$
	5  Rotation + 90 degrés	$T_{(u,v)} = (-1)^u B_{(v,u)}$
	6  Rotation -90 degrés	$T_{(u,v)} = (-1)^v B_{(v,u)}$
	7  Réflexion / 1 ^{re} diagonale	$T_{(u,v)} = B_{(v,u)}$
Transf. Massique	$T(x,y) = aB(x,y) + b$	$T_{(u,v)} = a B_{(u,v)} + bN \begin{matrix} 0 & 0 \\ \delta_u & \delta_v \end{matrix}$

Figure 3.6 : Transformations Ψ dans le domaine spatial et fréquentiel

En fait, quelle que soit l'isométrie considérée, elle implique seulement une permutation de u et v ou la multiplication de certains coefficients par -1 . Cette caractéristique justifie en partie l'utilisation de la *TC D* car la construction d'une représentation invariante à ces opérations va être possible.

3.1.3.4 Construction d'une représentation invariante

Pour accélérer l'étape de recherche du meilleur bloc \mathcal{D} , nous cherchons à construire une représentation canonique des blocs telle que chaque bloc ait un modèle invariant quelle que soit la transformation qui lui est appliquée.

La construction de l'invariance à une transformation massique permet de ne pas considérer des valeurs discrètes de a et b . Cela évite donc d'avoir à tester chaque bloc D avec ces valeurs et donc d'avoir à comparer chaque bloc I avec une série de versions transformées d'un bloc D comme c'est le cas dans l'algorithme mis au point par Jacquin (1992) où une trentaine de paires (a, b) sont considérées pour chaque bloc D candidat. L'invariance massique permet donc de réduire considérablement le temps de calcul par rapport à cette méthode.

Pour ce qui est de l'invariance aux transformations géométriques, elle va permettre de réduire le nombre de blocs candidats de $8 \times \lambda_{frac}$ à λ_{frac} .

1. Construction d'invariance massique sans perte :

Cette étape transforme un bloc \mathcal{B} en un bloc dit normalisé noté \mathcal{B}^{nor} . La déno-

mination *sans perte* signifie que cette étape ne génère aucune baisse d'optimalité par rapport au résultat que l'on obtiendrait en travaillant sur le bloc \mathcal{B} avant normalisation.

Par la suite, nous notons B_B le coefficient *DC* du bloc \mathcal{B} et A_B la norme \mathcal{L}^2 du bloc \mathcal{B} amputé du coefficient *DC* :

$$B_B = B(0,0). \quad (3.13)$$

$$(3.14)$$

$$A_B = \sqrt{\sum_{(u,v) \neq (0,0)}^{(N-1, V-1)} [B(u,v)]^2}. \quad (3.15)$$

- (a) Pour rendre le bloc invariant à un décalage b , il suffit de mettre le coefficient *DC* (c'est à dire l'intensité moyenne du bloc) à 0. En effet, l'équation 3.12 montre que le terme b affecte seulement le coefficient *DC* du bloc :

$$B^{nor}(0,0) = 0. \quad (3.16)$$

- (b) Pour rendre le bloc invariant au changement d'échelle a , il faut normaliser les autres coefficients. Nous utilisons une norme \mathcal{L}^2 . En effet, l'équation 3.12 montre que le terme a affecte uniformément tous les coefficients du bloc par un terme multiplicatif.

$$\mathcal{B}^{nor}(u, v) = \frac{1}{A_B} \times \mathcal{B}(u, v) \quad \forall (u, v) \neq (0, 0) . \quad (3.17)$$

(c) Expression des paramètres de la transformation massique :

Selon la méthode de Wohlberg et de Jager (1995), une fois trouvé le meilleur candidat \mathcal{D}^{nor} pour un bloc \mathcal{I}^{nor} donné, les paramètres de transformation massique sont déduits dans le domaine spatial. Ils suivent les expressions données à l'équation 3.18 :

$$b = \frac{B_I - aB_D}{N} \quad \text{et} \quad a = \frac{A_D}{A_I} . \quad (3.18)$$

où I et D sont les versions dans le domaine spatial original des blocs \mathcal{I}^{nor} et \mathcal{D}^{nor} .

2. Justification :

Nous avons vu que dans l'espace $TC\mathcal{D}$, le meilleur bloc \mathcal{D} satisfaisait le critère suivant :

$$E = \min_{\Psi, \lambda_{frac}} \sqrt{\sum_{(u,v)=(0,0)}^{(N-1,N-1)} \frac{1}{N^2} [\mathcal{I}(u, v) - \Psi \circ \mathcal{D}(u, v)]^2}$$

soit, en exprimant Ψ :

$$E = \min_{i, \lambda_{frac}} \sqrt{\sum_{(u,v)=(0,0)}^{(N-1,N-1)} \frac{1}{N^2} [\mathcal{I}(u, v) - a\mathcal{D}_i(u, v) - bN\delta_0^u\delta_0^v]^2}$$

où \mathcal{D}_i ($i = 0, \dots, 7$) représente le bloc \mathcal{D} transformé par une des 8 isométries possibles.

En remplaçant a et b par leurs expressions, nous obtenons :

$$\begin{aligned}
 E &= \min_{i, \lambda_{frac}} \sqrt{\sum_{(u,v) \neq (0,0)}^{(N-1, N-1)} \frac{1}{N^2} \left[\mathcal{I}(u, v) - \frac{A_D}{A_I} \mathcal{D}_i(u, v) - \frac{B_I - \frac{A_D}{A_I} B_D}{N} N \delta_0^u \delta_0^v \right]^2} \\
 E &= \frac{1}{N \times A_I} \min_{i, \lambda_{frac}} \sqrt{\sum_{(u,v) \neq (0,0)}^{(N-1, N-1)} [A_I \mathcal{I}(u, v) - A_D \mathcal{D}_i(u, v) - (B_I A_I - A_D B_D) \delta_0^u \delta_0^v]^2} \\
 E &= \frac{1}{N \times A_I} \min_{i, \lambda_{frac}} \sqrt{\sum_{(u,v) \neq (0,0)}^{(N-1, N-1)} [A_I (\mathcal{I}(u, v) - B_I \delta_0^u \delta_0^v) - A_D (\mathcal{D}_i(u, v) - B_D \delta_0^u \delta_0^v)]^2}
 \end{aligned}$$

Comme, par définition, $B_I = \mathcal{I}(0,0)$ et $B_D = \mathcal{D}(0,0)$, cette expression est équivalente à :

$$E = \min_{i, \lambda_{frac}} \sqrt{\sum_{(u,v) \neq (0,0)}^{(N-1, N-1)} [A_I \mathcal{I}(u, v) - A_D \mathcal{D}_i(u, v)]^2}$$

Soit, en reprenant les notations \mathcal{I}^{nor} et \mathcal{D}^{nor} , nous obtenons finalement l'équation 3.19 :

$$E = \min_{i, \lambda_{frac}} \sqrt{\sum_{(u,v) \neq (0,0)}^{(N-1, N-1)} [\mathcal{I}^{nor}(u, v) - \mathcal{D}_i^{nor}(u, v)]^2}. \quad (3.19)$$

Nous obtenons donc une équivalence parfaite entre minimiser l'erreur dans le domaine spatial ou fréquentiel avant ou après invariance massique. Cependant, à ce stade, il faut encore présenter au système les 8 isométries du bloc D pour choisir la meilleure approximation du bloc I . Nous choisissons alors d'effectuer

une construction d'invariance géométrique avec perte.

3. Construction d'invariance géométrique avec perte :

L'examen de la figure 3.6 permet de comprendre pourquoi nous menons la construction d'invariance géométrique avec perte en deux étapes. Comme décrit précédemment, les isométries peuvent affecter le bloc \mathcal{B}^{nor} de deux manières: soit en permutant les paramètres de position u et v , soit en modifiant les signes de certains coefficients.

(a) **Information de transposition :**

Nous pouvons réduire le temps de recherche de $8 \times \lambda_{frac}$ à $4 \times \lambda_{frac}$ en rendant le bloc \mathcal{B}^{nor} invariant aux transpositions c'est-à-dire aux permutations entre les coefficients u et v de $\mathcal{B}^{nor}(u, v)$.

Le résultat de cette étape est noté \mathcal{B}^{trans} .

Pour rendre le bloc invariant aux transpositions, nous fixons la contrainte suivante: l'amplitude du triangle inférieur A^{inf} du bloc \mathcal{B}^{trans} doit être inférieure à celle du triangle supérieur A^{sup} . Les amplitudes sont évaluées selon les équations suivantes :

$$A^{inf}(\mathcal{B}) = \sum_{v=0}^{N-1} \sum_{u=v+1}^{N-1} \mathcal{B}^{nor}(u, v). \quad (3.20)$$

$$A^{sup}(\mathcal{B}) = \sum_{u=0}^{N-1} \sum_{v=u+1}^{N-1} \mathcal{B}^{nor}(u, v). \quad (3.21)$$

Il est à noter que ces deux expressions excluent la diagonale du bloc. Un drapeau de permutation est déterminé tel que :

$$P_B = \begin{cases} 0 & \text{si } A^{inf}(\mathcal{B}) \leq A^{sup}(\mathcal{B}). \\ 1 & \text{sinon.} \end{cases} \quad (3.22)$$

Ainsi, nous obtenons:

$$\mathcal{B}^{trans}(u, v) = \begin{cases} \mathcal{B}^{nor}(u, v) & \text{si } P_B = 0. \\ \mathcal{B}^{nor}(v, u) & \text{sinon.} \end{cases} \quad (3.23)$$

La comparaison des drapeaux des blocs \mathcal{I}^{trans} et \mathcal{D}^{trans} fixe alors les isométries possibles numérotées d'après la figure 3.6 :

♣ Si $P_I = P_D$, les deux blocs \mathcal{I}^{nor} et \mathcal{D}^{nor} ont leurs triangles d'amplitude maximale placés à la même position. Les isométries i possibles pour approximer I par D_i sont donc 0, 1, 2 ou 3 car elles n'impliquent pas de permutation de u et v .

♣ Si $P_I \neq P_D$, les deux blocs \mathcal{I}^{nor} et \mathcal{D}^{nor} ont leurs triangles d'amplitude maximale en opposition. Les isométries i possibles pour approximer I par D_i sont 4, 5, 6 ou 7.

Nous devons remarquer que nous introduisons une perte d'optimalité. En effet, cette étape de transposition additionne des termes de signes opposés pour calculer l'amplitude des triangles A^{inf} et A^{sup} (voir équations 3.21).

De ce fait, certains coefficients importants en amplitude, mais de signes opposés peuvent se compenser mutuellement; ceci peut entraîner quelques choix d'isométrie non optimaux.

(b) **Construction d'une représentation canonique :**

Cette ultime étape réduit le temps de recherche de $4 \times \lambda_{frac}$ à λ_{frac} .

Pour rendre le bloc \mathcal{B}^{trans} invariant aux isométries, il faut le rendre insensible aux changements de signe de certains coefficients générés par ces isométries.

Les coefficients du bloc \mathcal{B}^{trans} peuvent être répartis en 3 catégories selon les valeurs de u et v . Ces trois catégories correspondent chacune à des sensibilités à certaines isométries. Le tableau 3.1 représente les différents cas possibles.

Tableau 3.1 : Classes des coefficients TCD

u	v	Sensibles à	Isométries correspondantes	Classe
impair	pair	$(-1)^u$ ou $(-1)^{u+v}$	2.3.4 et 5	1
pair	impair	$(-1)^v$ ou $(-1)^{u+v}$	1.3.4 et 6	2
impair	impair	$(-1)^u$ ou $(-1)^v$	1.2.5 et 6	3

Le cas " u et v pairs" n'est pas considéré car il n'est affecté du point de vue changement de signe par aucune isométrie.

Il existe une inter-corrélation de type OU entre les 3 classes.

Ceci signifie que chaque isométrie pouvant affecter une classe affecte simultanément également une (et une seule) des deux autres classes.

Ainsi, la classe 3 est corréliée avec la classe 1 ou la classe 2 à cause de la parité des indices u et v qui les définissent. Par exemple, si une isométrie impliquant la transformation contenant le terme $(-1)^u$ est appliquée au bloc, les classes 1 et 3 sont affectées, mais pas la classe 2. Pour que la classe 3 ne soit pas affectée, il faut appliquer en plus une transformation contenant le terme $(-1)^v$ (ce qui donne globalement le terme $(-1)^{u+v}$) ce qui laisse la classe 1 affectée mais affecte par la même occasion la classe 2. En conclusion, vu la corrélation existant entre les 3 classes, on ne peut donc pas appliquer une isométrie au bloc dont la transformation affectera une classe sans en affecter une des deux autres.

Nous appliquons un masque sur le bloc \mathcal{B}^{trans} qui va consister à prendre la valeur absolue des coefficients dont le signe peut être modifié par une ou plusieurs des isométries possibles c'est-à-dire tous les termes du bloc $\mathcal{B}^{trans}(u, v)$ pour lesquels u ou bien v est impair. Le bloc résultant de l'invariance canonique appliquée à \mathcal{B}^{trans} est noté \mathcal{B}^{can} et a pour expression :

$$\mathcal{B}^{can}(u, v) = \begin{cases} |\mathcal{B}^{trans}(u, v)| & \text{si } u \text{ ou bien } v \text{ est impair.} \\ \mathcal{B}^{trans}(u, v) & \text{sinon.} \end{cases} \quad (3.24)$$

Le masque appliqué sur la matrice TCD du bloc est représenté à la figure 3.7 qui résume les différentes étapes de la construction globale de la

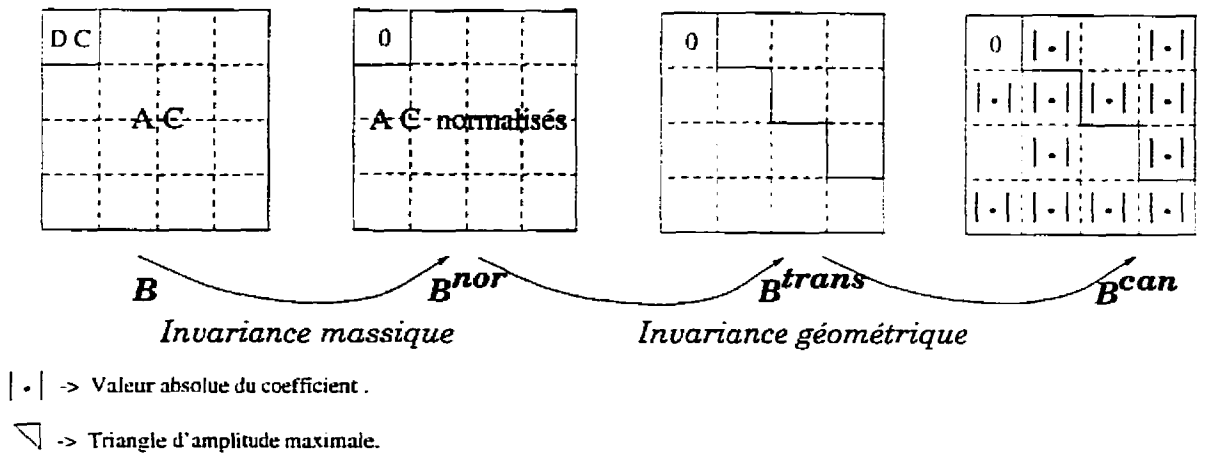


Figure 3.7 : Représentation canonique d'un bloc

représentation canonique d'un bloc.

Finalement, le meilleur bloc \mathcal{D}^{can} satisfait le critère suivant :

$$E = \min_{\lambda_{frac}} \sqrt{\sum_{(u,v) \neq (0,0)}^{(N-1, N-1)} [\mathcal{I}^{can}(u, v) - \mathcal{D}^{can}(u, v)]^2}. \quad (3.25)$$

(c) Conséquences de la construction invariante :

Il faut noter que l'utilisation de ce masque est source d'erreurs importantes car la perte de l'information de signe concerne les trois-quarts des coefficients du bloc \mathcal{B}^{trans} . Or, ceci peut s'avérer dramatique lorsque l'information de signe concerne des termes d'amplitude élevée en valeur absolue du bloc I à coder. À ce stade, nous introduisons la notion de terme *prépondérant*.

Définition : Nous appelons terme *prépondérant* un terme, positif ou négatif, dont l'amplitude en valeur absolue est suffisamment élevée pour nécessiter une bonne approximation de ce terme.

Exemple catastrophique ! La figure 3.8 montre un cas pour lequel la construction canonique a mené au choix d'un bloc domaine très mauvais.

Nous avons représenté pour chaque bloc les termes prépondérants du bloc \mathcal{I}^{trans} . Les autres termes non inscrits n'ont que très peu d'influence.

Les blocs \mathcal{D}_1^{trans} et \mathcal{D}_2^{trans} sont deux candidats appartenant à la sirale de recherche de longueur λ_{rac} pour approximer le bloc \mathcal{I}^{trans} . La représentation canonique avec perte (invariance géométrique) mène à choisir \mathcal{D}_2^{can} comme meilleur bloc pour approximer \mathcal{I}^{can} . En effet, l'erreur quadratique moyenne entre ces deux blocs est beaucoup plus faible qu'entre \mathcal{D}_1^{can} et \mathcal{I}^{can} . Or, en calculant après coup le meilleur bloc transformé par isométrie $iso_2(\mathcal{D}_1^{trans})$ et $iso_3(\mathcal{D}_2^{trans})$, nous apercevons que, en recouvrant l'information de signe, $iso_2(\mathcal{D}_1^{trans})$ constitue *en réalité* le meilleur bloc pour approximer \mathcal{I}^{trans} .

Explication : Dans l'exemple précédent, il s'avère que les termes prépondérants du bloc \mathcal{I}^{trans} se répartissent dans les trois classes définies dans le tableau 3.1. Par conséquent, ce sont des termes dont le signe peut être

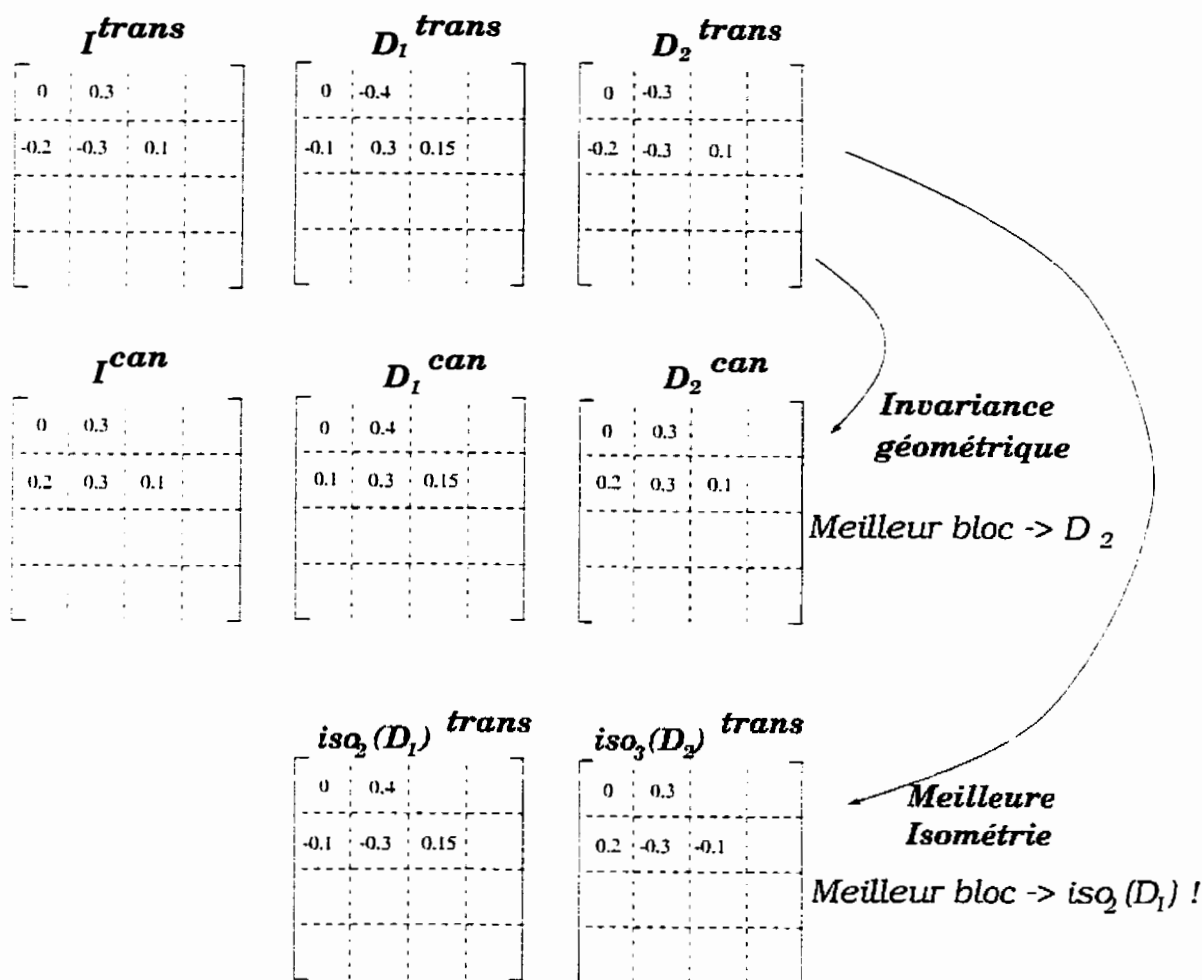


Figure 3.8 : Exemple pathologique de la représentation invariante

modifié par une isométrie. Or, si l'on observe attentivement le tableau 3.1, nous nous rendons compte que chaque isométrie apparait dans *deux* classes. Les changements de signe des termes de chaque classe sont donc corrélés. De ce fait, il peut être impossible de réattribuer le bon signe à tous ces termes par l'application globale d'une isométrie sur le bloc.

Ainsi, dans l'exemple, le terme $\mathcal{D}_2^{trans}(0, 1)$ appartient à la classe 1 et voit son signe modifié sous l'application sur le bloc de l'isométrie 3. Cette isométrie contient la transformation $(-1)^{u+v}$ et affecte donc les termes du bloc pour lesquels u ou v est impair. Donc, nous pouvons voir que les termes $\mathcal{D}_2^{trans}(1, 0)$ et $\mathcal{D}_2^{trans}(1, 2)$ tous deux de la classe 2 sont aussi affectés par cette isométrie, mais pas le terme $\mathcal{D}_1^{trans}(1, 1)$ de la classe 3.

Il n'a donc été possible que de réattribuer les bons signes (ceux de chaque terme prépondérant de \mathcal{I}^{trans}) à deux termes sur les quatre de $iso_3(\mathcal{D}_2^{trans})$ ce qui résulte en une erreur quadratique moyenne importante.

Pour éviter ces erreurs de choix, il est donc nécessaire d'effectuer une étude minutieuse des signes des termes prépondérants de \mathcal{I}^{trans} et des termes correspondants dans chaque bloc domaine candidat \mathcal{D}^{trans} . La méthode d'étude de compatibilité de ces signes vise à rejeter tous les blocs domaines candidats \mathcal{D}^{trans} déclarés incompatibles avec le bloc \mathcal{I}^{trans} à coder et est décrite dans le paragraphe suivant.

3.1.3.5 Etude des impacts du masque

Cette étude consiste à décider si, pour un bloc \mathcal{I}^{trans} à coder, la perte d'information de signe sur les termes prépondérants de ce bloc placés sur le masque (c'est-à-dire pour lesquels u ou v sont impairs) va nécessiter une étude de compatibilité de signes avec un bloc domaine candidat \mathcal{D}^{trans} et son éventuel rejet.

Un exemple détaillé à la fin de ce paragraphe permet de mieux comprendre les différentes étapes de cette étude.

L'annexe C contient toutes les explications et les détails algorithmiques de cette section.

Dans cette étude, deux paramètres fixes vont être considérés :

- Γ_{prep} , le seuil en amplitude au-dessus duquel un terme d'un bloc est considéré comme prépondérant;
- Δ_{prep}^{max} , le nombre maximum de termes prépondérants considérés dans chaque bloc \mathcal{I}^{trans} .

L'étude de compatibilité des blocs est présentée à la figure 3.9. Conformément à l'organigramme, l'étude de impacts du masque comporte les éléments suivants :

1. Calcul des positions des termes prépondérants sur le masque :

Le but de cette étape est d'évaluer si les coefficients placés sur le masque du bloc \mathcal{I}^{can} comportent des termes prépondérants et le cas échéant leur position.

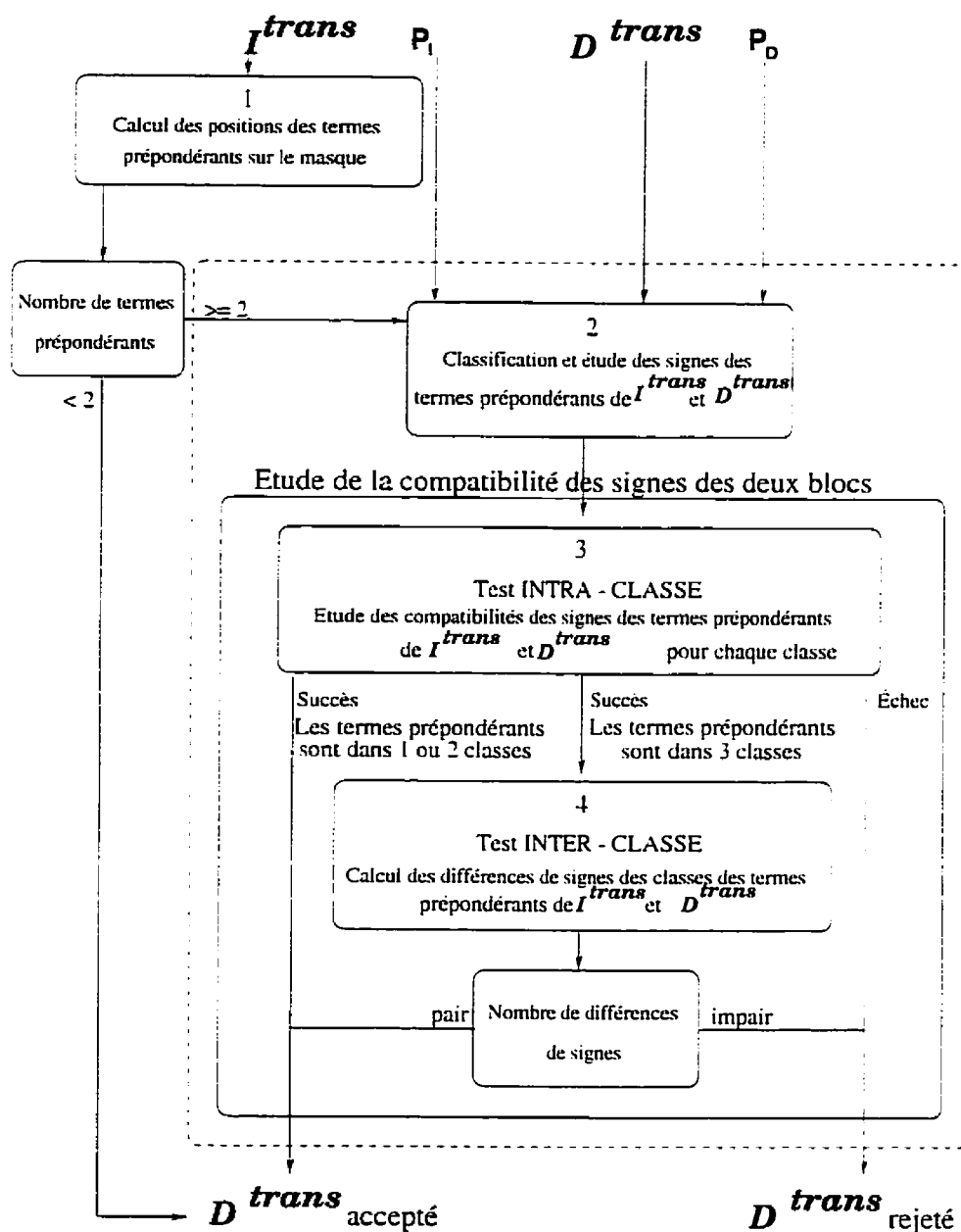


Figure 3.9 : Organigramme de l'étude de compatibilité des blocs

2. Classification et étude des signes des termes prépondérants

Lorsque le nombre de termes prépondérants est supérieur ou égal à 2, il est nécessaire d'examiner les classes et les signes de ces termes dans le bloc \mathcal{I}^{trans} correspondant et dans chaque bloc \mathcal{D}^{trans} candidat. Ceci va nous permettre de savoir si une isométrie sera ensuite capable de réattribuer le bon signe (celui de chaque terme prépondérant de \mathcal{I}^{trans}) à chaque terme correspondant dans le bloc \mathcal{D}^{trans} candidat. Si ce n'est pas le cas, le bloc \mathcal{D}^{trans} sera rejeté.

3. Compatibilité des signes des blocs \mathcal{I}^{trans} et \mathcal{D}^{trans}

Les incompatibilités de signes des termes prépondérants du bloc \mathcal{I}^{trans} avec \mathcal{D}^{trans} dépendent de la classe à laquelle appartiennent ces termes.

En effet, les classes déterminent les corrélations existant entre ces termes. Cette étude sert à rejeter les blocs \mathcal{D}^{trans} candidats dont l'erreur résultant entre \mathcal{D}^{can} et \mathcal{I}^{can} ne reflèterait pas celle entre \mathcal{D} et \mathcal{I} . Ceci permet d'éviter des calculs inutiles et des résultats trompeurs.

Deux types de tests vont être utilisés :

(a) un test dit **INTRA-CLASSE** :

Il est mené de manière indépendante sur chacune des trois classes. Il consiste à étudier les compatibilités des signes de tous les termes prépondérants des blocs \mathcal{D}^{trans} et \mathcal{I}^{trans} appartenant à une même classe.

(b) un test dit **INTER-CLASSES** :

Il consiste à comparer les compatibilités de signes des classes des deux blocs entre elles. Il n'a lieu d'être que lorsque les termes prépondérants de \mathcal{I}^{trans} se répartissent dans les trois classes possibles.

Comme nous pouvons le constater sur l'organigramme 3.9, deux cas peuvent se présenter :

(a) les termes prépondérants se répartissent dans 1 ou 2 classes différentes :

Dans le cas de 2 classes, celles-ci ne sont pas corrélées car il est toujours possible de trouver une isométrie affectant l'une mais pas l'autre.

Seul le test *intra-classe* doit être effectué. S'il est concluant, le bloc \mathcal{D}^{trans} est classé comme compatible avec le bloc à coder \mathcal{I}^{trans} .

(b) les termes prépondérants se répartissent dans les 3 classes :

Les deux tests *intra-classe* et *inter-classes* doivent être effectués et concluants pour que le bloc \mathcal{D}^{trans} soit classé comme compatible avec le bloc à coder \mathcal{I}^{trans} .

4. Exemple :

La figure 3.10 montre comment l'étude de compatibilité de signes permet de rejeter le bloc menant à un mauvais résultat dans l'exemple présenté sur la figure 3.8. Le détail des calculs relatif à cette étude est présenté dans l'annexe C.

Considérant le bloc courant à coder \mathcal{I}^{trans} , nous voulons déterminer si deux blocs domaines candidats \mathcal{D}_1^{trans} et \mathcal{D}_2^{trans} sont compatibles avec \mathcal{I}^{trans} .

L'étape du calcul du nombre de termes prépondérants du bloc \mathcal{I}^{trans} montre qu'il comporte 4 termes prépondérants positionnés comme montré à la figure 3.10.

Comme ce nombre est supérieur à 2, il est nécessaire d'effectuer l'étude de la compatibilité des signes.

L'étape suivante consiste à calculer les classes des termes prépondérants du bloc \mathcal{I}^{trans} en suivant le tableau 3.1. Ensuite, pour les trois blocs \mathcal{I}^{trans} , \mathcal{D}_1^{trans} et \mathcal{D}_2^{trans} candidats, nous calculons les signes des termes prépondérants de \mathcal{I}^{trans} .

Les quatre termes prépondérants sont de trois classes différentes. Il faut donc mener en premier lieu l'étude de compatibilité de signes intra-classe. Celle-ci réussit pour les deux blocs.

Finalement, la dernière étape d'étude de la compatibilité de signes inter-classes amène à accepter \mathcal{D}_1^{trans} et à rejeter \mathcal{D}_2^{trans} .

5. Nouveau critère de succès :

Pour un bloc \mathcal{I}^{can} tel que $E_I = 1$, le meilleur bloc \mathcal{D}^{can} satisfait le critère suivant :

$$E = \min_{\lambda_{comp, frac}} \sqrt{\sum_{(u,v) \neq (0,0)}^{(N-1, N-1)} [\mathcal{I}^{can}(u, v) - \mathcal{D}^{can}(u, v)]^2}. \quad (3.26)$$

I^{trans}	D_1^{trans}	D_2^{trans}
0 0.3	0 -0.4	0 -0.3
-0.2 -0.3 0.1	-0.1 0.3 0.15	-0.2 -0.3 0.1

a) 4 termes prépondérants de classe:

	1	
2	3	2

b) Etude des signes des termes prépondérants

	+	
-	-	+

	-	
-	+	+

	-	
-	-	+

c) Compatibilité des signes intra-classe:

1	2	3
+	+	+

-	+	-
---	---	---

-	+	+
---	---	---

c) Nombre de différences de signes

2 -> D_1 compatible avec I 1 -> D_2 incompatible avec I

ACCEPTÉ

REJETÉ

Figure 3.10 : Exemple d'étude de compatibilité de signes

où $\lambda_{comp,frac}$ représente la restriction de λ_{frac} aux positions de la spirale pour lesquelles le bloc \mathcal{D}^{trans} a été jugé compatible avec \mathcal{I}^{trans} .

3.1.3.6 Expression des paramètres de la transformation

Une fois trouvé le meilleur candidat \mathcal{D}^{can} compatible pour un bloc \mathcal{I}^{can} donné, les paramètres de transformation massique sont déduits dans le domaine spatial et envoyés dans le canal en utilisant les résultats obtenus selon l'équation 3.18. En outre, ces valeurs réelles sont quantifiées en utilisant un quantificateur uniforme.

Ensuite, le système calcule dans le domaine spatial la meilleure transformation géométrique de D qui approxime I . En notant D_i le bloc D transformé par l'isométrie $i = 0 \dots 7$, le meilleur D_i satisfait :

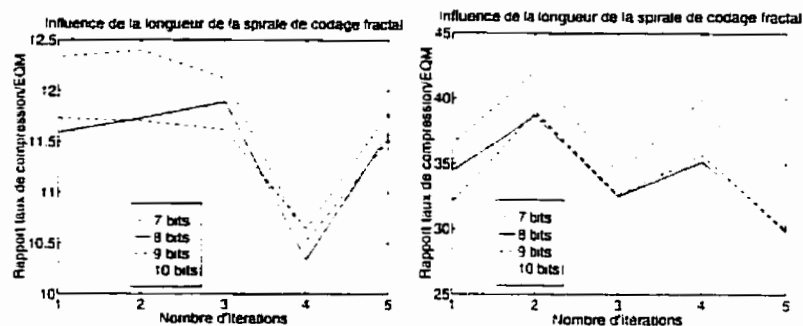
$$E = \min_i \sqrt{\sum_{(x,y)=(0,0)}^{(N-1,N-1)} \frac{1}{N^2} [I(x,y) - aD_i(x,y) - b]^2}. \quad (3.27)$$

Nous avons étudié les performances lorsque le masque est appliqué en tenant compte ou non des compatibilités de signe. Ces comparaisons sont décrites dans la prochaine section, ainsi que les influences de tous les autres paramètres intervenant dans le module de codage fractal: λ_{frac} , a et b .

3.1.3.7 Étude de l'influence des seuils et paramètres

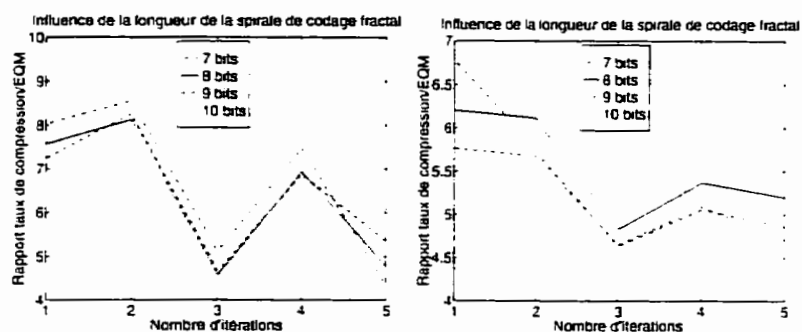
1. Influence de la longueur de la spirale de recherche λ_{frac} :

Les résultats présentés à la figure 3.11 (où λ_{frac} est donné en nombre de bits)



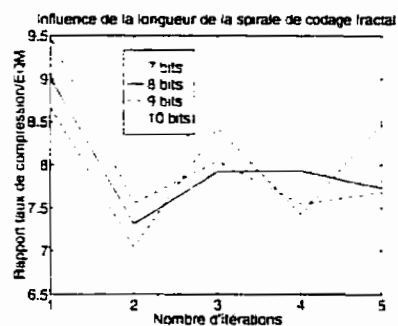
(a) Rubic

(b) Taxi



(c) NASA

(d) Papillon



(e) Tunnel

Figure 3.11 : Influence de la longueur de la spirale de recherche λ_{frac} sur le rapport entre le taux de compression et l'erreur quadratique moyenne

varient selon les séquences. La figure D.6 montre que, pour toutes les séquences tests, il faut rejeter le cas $\lambda_{frac} = 128$ car ceci génère une erreur quadratique moyenne trop importante sur l'image reconstruite car le domaine de recherche est trop rétréci.

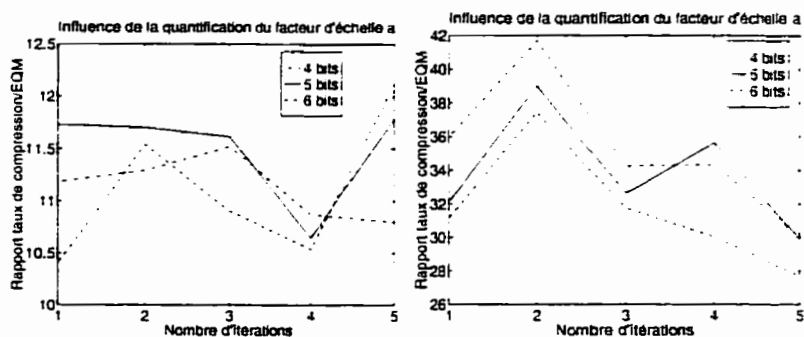
La séquence "Tunnel" présente son aspect optimal pour la valeur intermédiaire $\lambda_{frac} = 512$. Pour les séquences "Taxi" et "Rubic", il faut rejeter $\lambda_{frac} = 1024$ car, malgré une erreur faible EQM , la compression résultante est trop diminuée. Pourtant, pour les séquences "NASA" et "Papillon", ce cas donne de bons résultats. Ceci s'explique par le fait que ces deux séquences comportent un mouvement complexe et utilisent donc plus le codage fractal. Plus le domaine de recherche est grand, meilleurs sont les résultats. Par contre, pour les autres séquences, l'ajout de bit induit par l'augmentation de λ_{frac} n'est pas justifié.

Bizarrement, la séquence "Papillon" donne encore de bons résultats pour $\lambda_{frac} = 256$, mais pas pour $\lambda_{frac} = 512$. En fait, $\lambda_{frac} = 256$ correspond à un cas où la compression est bonne car la compensation du mouvement pallie à la faiblesse du module fractal, cependant pour une erreur quadratique moyenne médiocre: $\lambda_{frac} = 512$ est un cas intermédiaire qui sera plus satisfaisant visuellement.

Nous pensons rejeter les deux extrêmes car ils risquent d'être très mauvais sur certaines séquences. En fait, pour obtenir des erreurs EQM faibles, il faudrait choisir $\lambda_{frac} = 512$ et pour favoriser une forte compression $\lambda_{frac} = 256$.

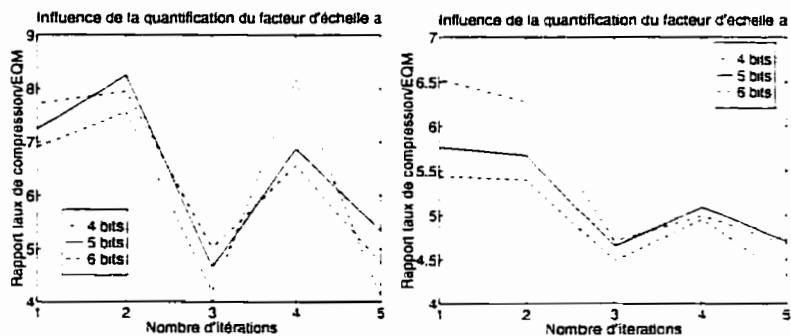
2. Influence de la quantification du facteur d'échelle a :

Les résultats présentés sur les figures 3.12, D.7 et D.8 montrent que toutes



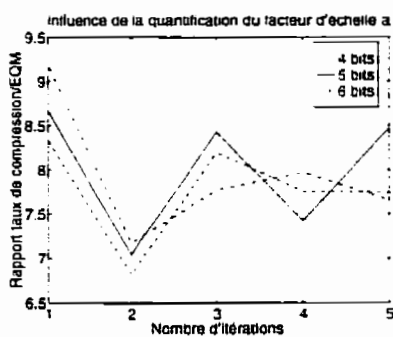
(a) Rubic

(b) Taxi



(c) NASA

(d) Papillon



(e) Tunnel

Figure 3.12 : Influence de la quantification du facteur d'échelle a sur le rapport entre le taux de compression et l'erreur quadratique moyenne

les séquences rejettent une quantification de a sur 6 bits (donc pour 64 valeurs possibles de a) car ceci diminue inutilement la compression. Nous choisissons de coder a sur 5 bits car c'est la valeur qui évolue le mieux après 5 itérations.

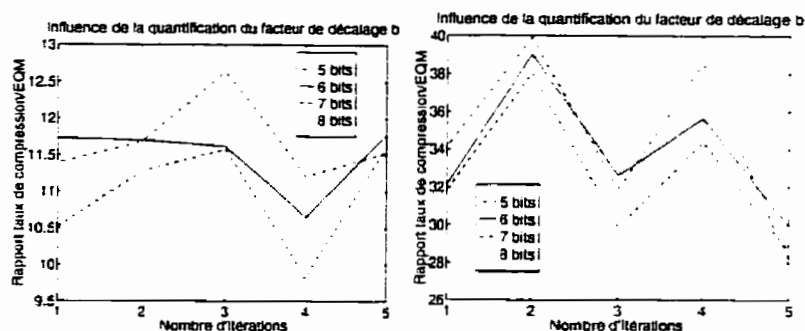
3. Influence de la quantification du facteur de décalage b :

Les résultats présentés sur les figures 3.13 , D.9 et D.10 montrent que toutes les séquences rejettent une quantification de b sur 8 bits car ceci diminue inutilement la compression. De même, prendre une quantification de b sur 4 bits résulte en une erreur quadratique moyenne trop élevée. Nous choisissons de coder b sur 6 bits car c'est la valeur qui évolue le mieux après 5 itérations.

4. Influence des paramètres d'étude de compatibilité des signes Γ_{prep} et Δ_{prep}^{max} :

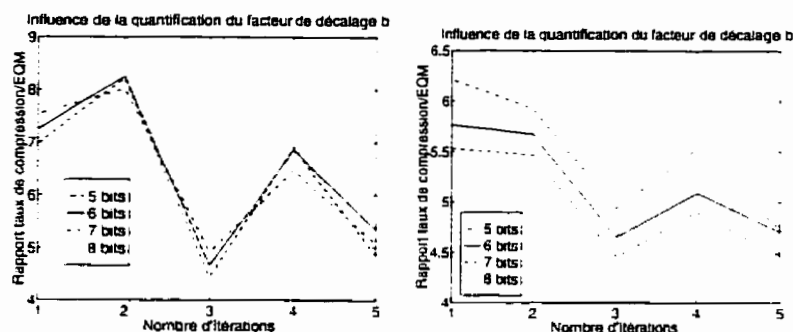
Les tests ont été menés en n'effectuant le codage fractal que pour le niveau de segmentation le plus fin avec des blocs de taille 4×4 footnotepour des blocs plus gros. le module de codage fractal est inhibé. Sur la figure 3.14. nous présentons les résultats obtenus pour plusieurs paires $(\Delta_{prep}^{max}, \Gamma_{prep})$. En effet, ces deux paramètres étant liés, il vaut mieux les considérer ensemble.

La courbe correspondant à " $(\Delta_{prep}^{max} = 0, \Gamma_{prep} = 0.14)$ " montre le résultat obtenu lorsqu'il n'y a aucune incompatibilité de signe. De ce fait, le module de codage fractal échoue très souvent en appariant des blocs très éloignés spatialement. La compression est importante (car le codage est fait par compensation du mouvement), mais l'erreur quadratique moyenne est très mauvaise. Pour la séquence "Taxi", nous voyons à la figure D.11 l'évolution dramatique de cette courbe:



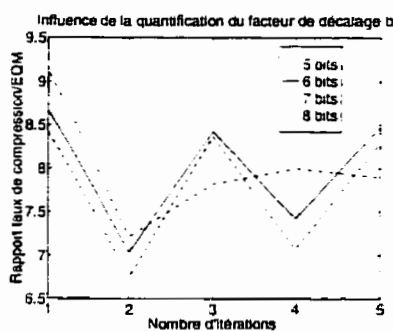
(a) Rubic

(b) Taxi



(c) NASA

(d) Papillon



(e) Tunnel

Figure 3.13 : Influence de la quantification du facteur de décalage b sur le rapport entre le taux de compression et l'erreur quadratique moyenne

la compression chute à cause d'une accumulation d'erreur due à trop de compensation du mouvement. Le système essaie donc de compenser en considérant plus de blocs actifs et plus de codage fractal: mais comme celui-ci est inefficace, l'erreur quadratique moyenne ne s'améliore pas au fil des itérations.

La courbe " $(\Delta_{prep}^{max} = 10, \Gamma_{prep} = 0.05)$ " reflète le cas inverse où une majorité de blocs vont être considérés comme incompatibles car, même des coefficients de faible amplitude (et donc de faible contribution au bloc) sont considérés comme prépondérants. Ainsi, tout comme le cas précédent, ceci résulte en un échec du module de codage fractal (qui est trop sévère), et donc en une erreur quadratique moyenne et une compression plus grandes.

Les trois autres courbes " $(\Delta_{prep}^{max} = 10, \Gamma_{prep} = 0.09)$ ", " $(\Delta_{prep}^{max} = 5, \Gamma_{prep} = 0.09)$ " et " $(\Delta_{prep}^{max} = 5, \Gamma_{prep} = 0.14)$ " se suivent beaucoup et montrent des cas intermédiaires. Ainsi, la courbe intermédiaire " $(\Delta_{prep}^{max} = 10, \Gamma_{prep} = 0.09)$ " correspond à un module fractal comportant une étude de compatibilité raisonnablement sévère ^{footnote}est-à-dire rejetant beaucoup de blocs domaines candidats. De même, " $(\Delta_{prep}^{max} = 5, \Gamma_{prep} = 0.14)$ " correspond à un module fractal comportant une étude de compatibilité raisonnablement laxiste ^{footnote}est-à-dire acceptant beaucoup de blocs domaines candidats.

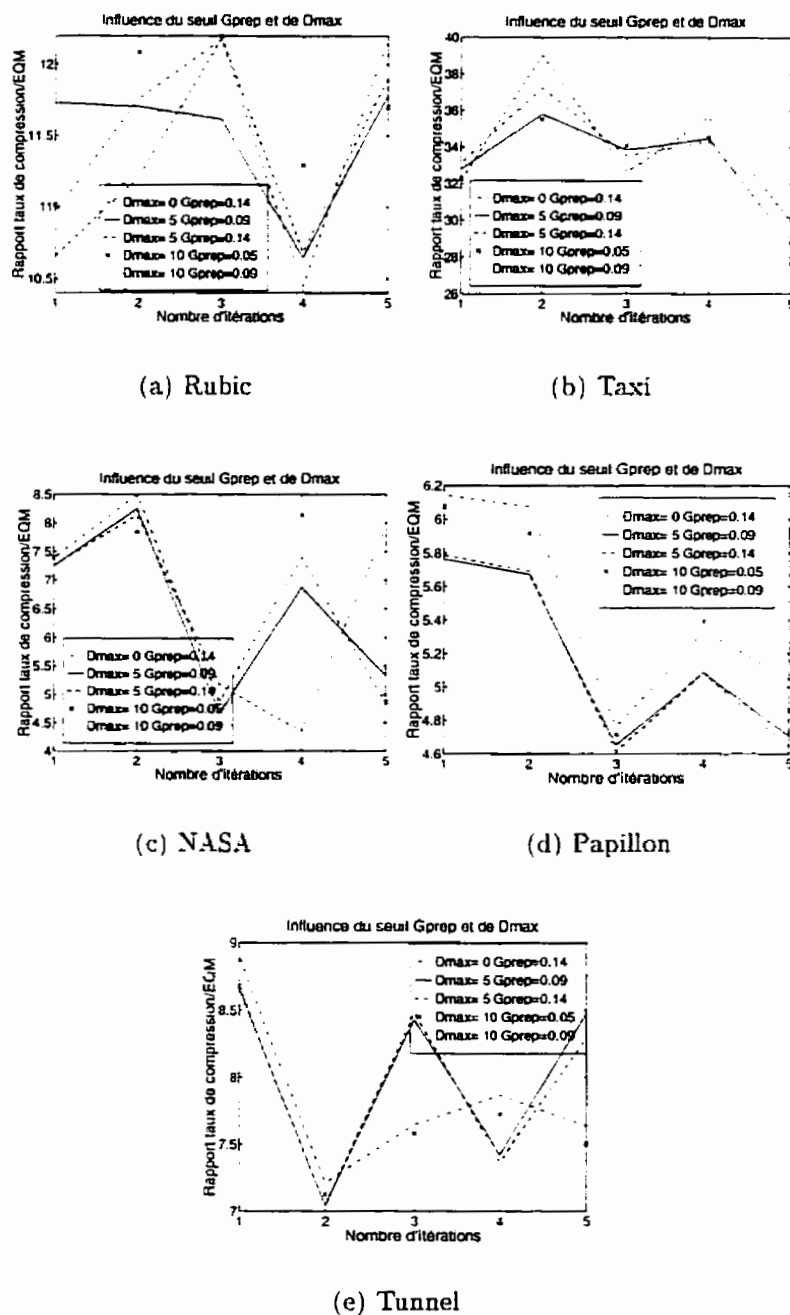


Figure 3.14 : Influence des paramètres d'étude de compatibilité des signes Γ_{prep} et Δ_{prep}^{max} sur le rapport entre le taux de compression et l'erreur quadratique moyenne

3.1.4 Module de segmentation en arbre quaternaire

La figure 3.15 illustre le module de segmentation en arbre quaternaire. Il permet de considérer des blocs de taille variable et donc de s'adapter à la complexité du contenu de l'image ou du mouvement dans la scène selon un mode "grossier à fin". Le critère utilisé pour diviser un bloc en 4 sous-blocs est l'échec successif des différents modules d'estimation des blocs actifs, de compensation du mouvement et de codage fractal dû à une erreur quadratique moyenne trop importante à chacun de ces modules.

3.1.4.1 Effet de la segmentation sur les différents modules

Nous avons étudié l'utilité d'effectuer une segmentation en arbre quaternaire sur chacun des trois modules de codage de notre système d'estimation des blocs actifs, de compensation du mouvement et de codage fractal.

Dans les deux expériences qui suivent, le module de codage fractal n'est mené qu'au niveau le plus fin (blocs 4×4).

1. *Influence de la segmentation en arbre quaternaire des blocs actifs :*

Cette expérience permet de voir s'il vaut mieux effectuer un codage multi-niveaux global ou seulement au niveau le plus fin avec des blocs de taille 4×4 .

Les résultats présentés à la figure 3.16 montrent que, pour toutes les séquences, il est profitable d'effectuer un codage multi-niveaux car, même si ceci peut entraîner une hausse de l'erreur quadratique moyenne globale, le gain en taux de

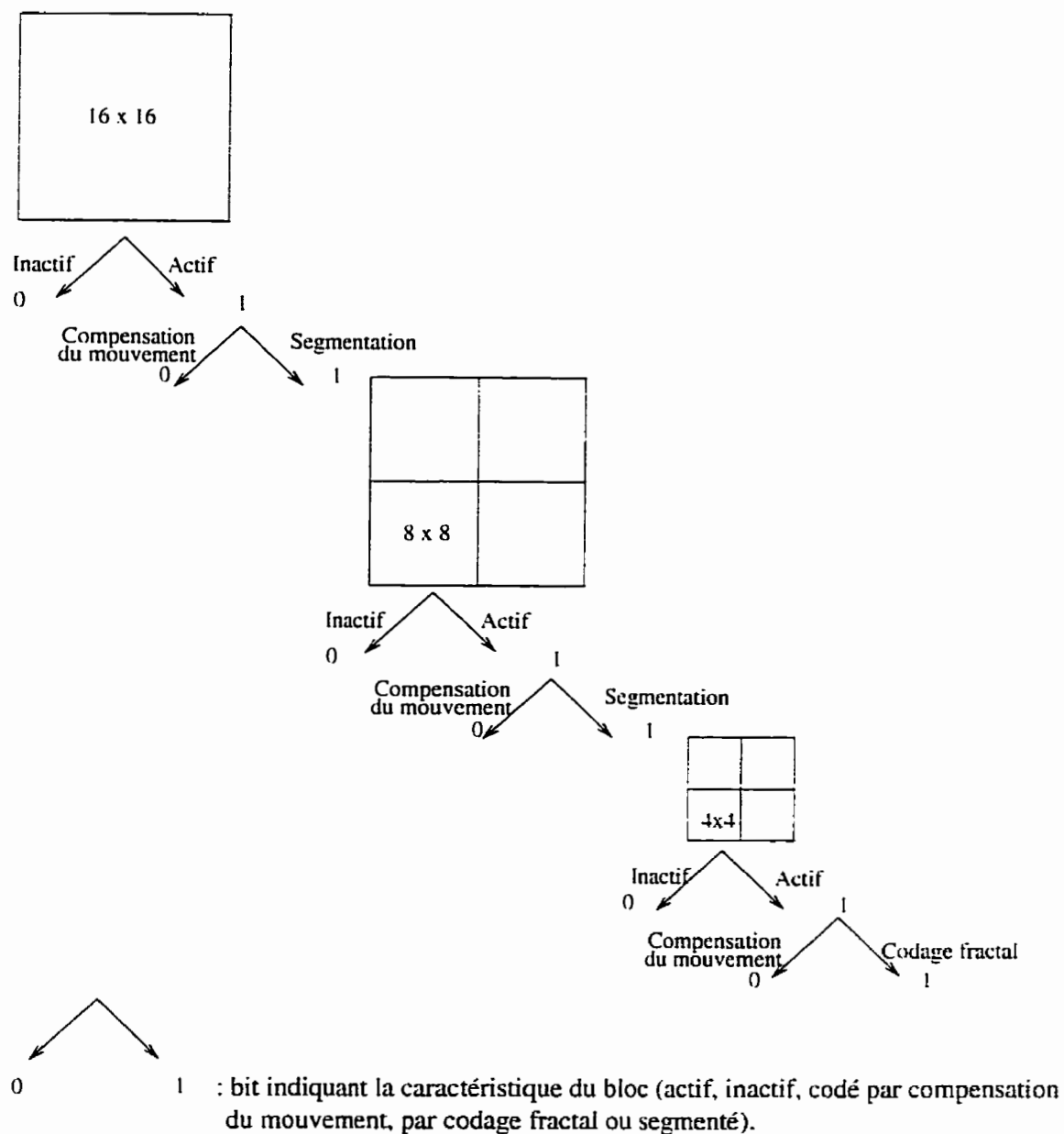
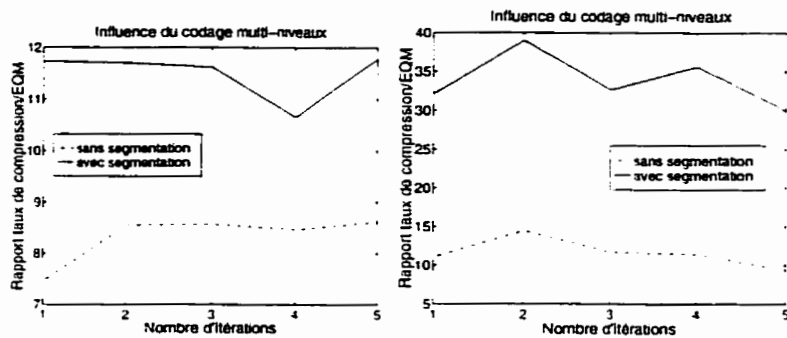
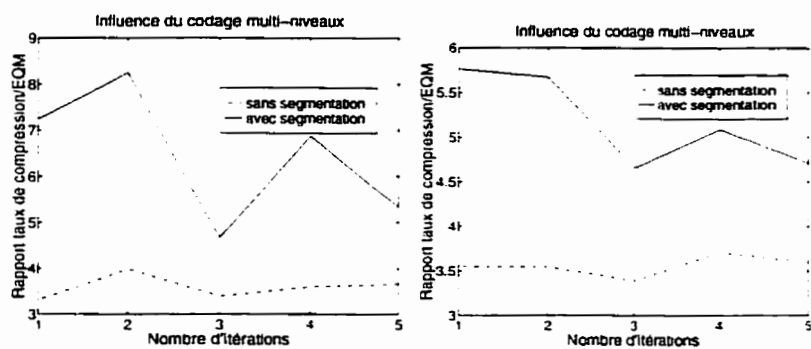


Figure 3.15 : Principe de segmentation en arbre quaternaire



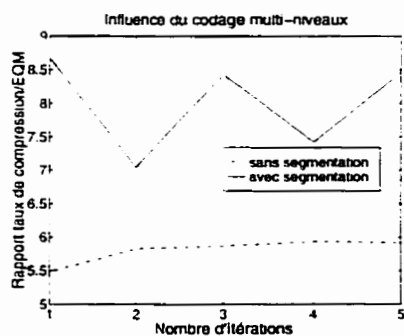
(a) Rubic

(b) Taxi



(c) NASA

(d) Papillon



(e) Tunnel

Figure 3.16 : Influence de la segmentation en arbre quaternaire des blocs actifs sur le rapport entre le taux de compression et l'erreur quadratique moyenne

compression est énorme.

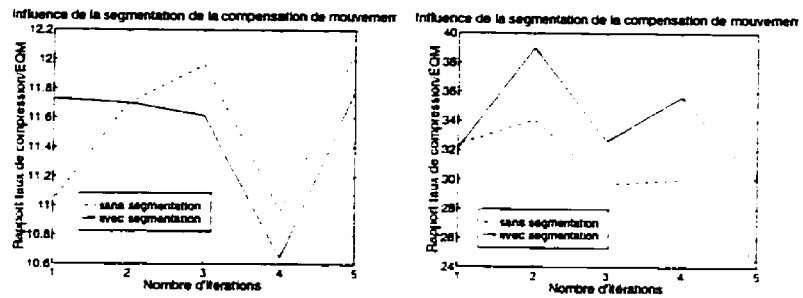
2. *Influence de la segmentation en arbre quaternaire des blocs actifs non estimables par compensation du mouvement :*

Cette expérience permet de voir si un bloc actif de niveau grossier doit passer dans le module de compensation du mouvement ou doit être directement segmenté au niveau de segmentation plus fin.

Nous avons mené deux séries d'expériences avec des valeurs de paramètres différentes.

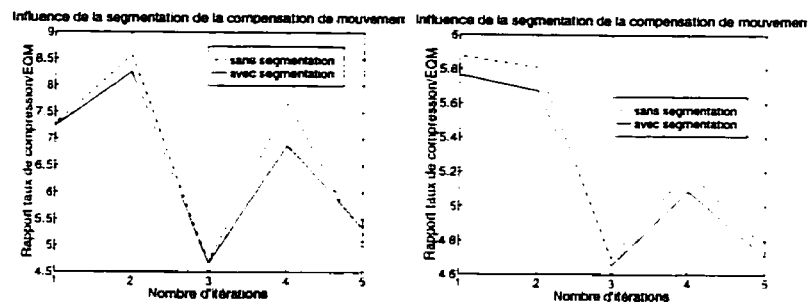
Pour la première expérience représentée à la figure 3.17, nous avons pris les valeurs des paramètres usuels $\Gamma_{mnt} = 6$ et $\lambda_{mnt} = 31$. Il s'avère que, seules les séquences "Taxi" et "Tunnel" sont favorables à un module de compensation du mouvement multi-niveaux. La séquence "NASA" devient favorable à un codage par compensation du mouvement multi-niveaux à la fin des 5 itérations. Les deux autres séquences "Rubic" et "Papillon" révèlent son inutilité. Face à ces résultats, nous avons décidé de tenter une deuxième expérience.

La deuxième expérience représentée à la figure 3.18 a consisté à réajuster $\Gamma_{mnt} = 8$ et de prendre $\lambda_{mnt} = 63$ pour favoriser les gros blocs. De plus, les valeurs de ces seuils ont donné de bons résultats lors de leur étude. Dans ce cas, pour toutes les séquences, un module de compensation du mouvement multi-niveaux s'avère toujours préférable à un module réservé aux blocs de taille 4×4 .



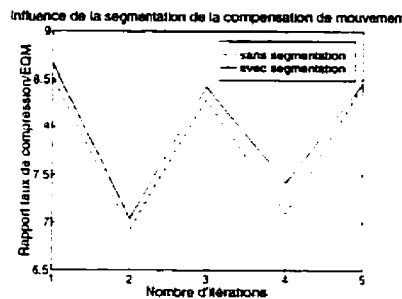
(a) Rubic

(b) Taxi



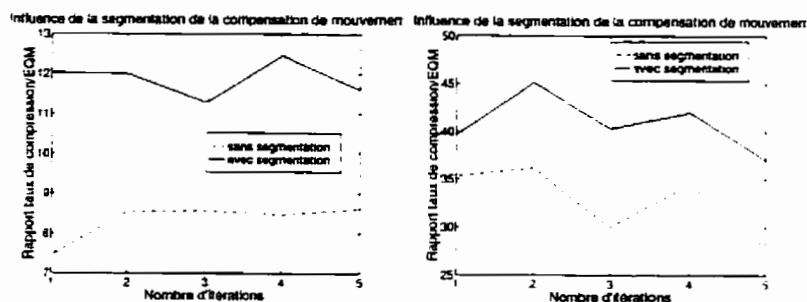
(c) NASA

(d) Papillon



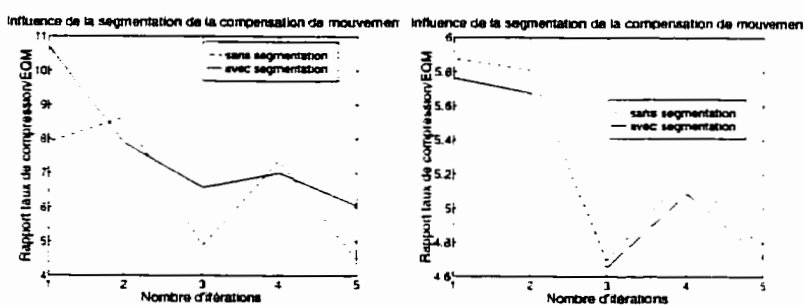
(e) Tunnel

Figure 3.17 : Influence de la segmentation en arbre quaternaire des blocs actifs non estimables par compensation du mouvement sur le rapport entre le taux de compression et l'erreur quadratique moyenne 1



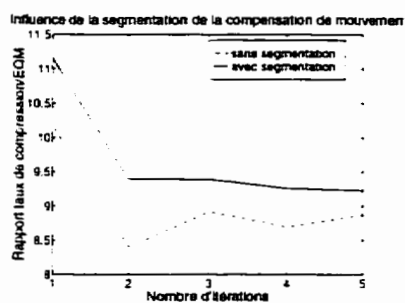
(a) Rubic

(b) Taxi



(c) NASA

(d) Papillon



(e) Tunnel

Figure 3.18 : Influence de la segmentation en arbre quaternaire des blocs actifs non estimables par compensation du mouvement sur le rapport entre le taux de compression et l'erreur quadratique moyenne 2

3. *Influence de la segmentation en arbre quaternaire du codage fractal :*

Pour toutes les séquences tests, nous avons observé que le système descendait toujours au niveau de segmentation le plus petit. Donc, nous n'effectuons de codage fractal qu'à ce niveau-là: c'est-à-dire pour des blocs images I de taille 4×4 .

3.2 Codage de la séquence à faible résolution L

Notre méthode, qui utilise la théorie de la neutralisation décrite dans la section 1.3 présente l'originalité d'être asymétrique pour les deux images. Cette asymétrie consiste à coder une des deux images H_t à une résolution plus grande que l'image homologue L_t .

L'image H_t à pleine résolution est codée à partir de l'image précédente reconstruite \hat{H}_{t-1} en utilisant la méthode décrite précédemment.

Son homologue stéréoscopique L_t est d'abord filtrée et sous-échantillonnée en utilisant une pyramide gaussienne dont nous avons adapté l'algorithme général décrit à la section A.3.1.

Ensuite, deux méthodes sont possibles pour encoder l'image résultante L_t^{fil} :

1. L_t^{fil} peut être codée exactement de la même manière que H_t :
2. ou L_t^{fil} peut être codée sensiblement de la même manière que H_t mais en remplaçant le module de compensation du mouvement par un module de compensation de disparité. Ce module cherche à apparier un bloc de L_t^{fil} en utilisant un bloc de l'image reconstruite filtrée de \hat{H}_t^{fil} placé sur la même bande épipolaire (normalement horizontale).

La théorie de la neutralisation est exploitée par la diminution de la résolution de L_t et par le fait que les seuils d'erreurs tolérables sont plus élevés pour coder l'image L_t à faible résolution que pour H_t .

La figure 3.19 illustre la théorie de la neutralisation pour une paire d'images stéréoscopiques d'une vue aérienne. Bien que l'image de droite n'est définie qu'en certaines régions éparses riches en information, le cerveau parvient à fusionner et à reconnaître les éléments principaux de la scène (voiture, maison).



(a) Image de gauche (b) Image de droite

Figure 3.19 : Illustration de la théorie de la neutralisation

Les deux sections qui suivent décrivent la méthode de filtrage et le module de compensation de disparité que nous avons développé.

3.2.1 Pyramide gaussienne

Nous rappelons qu'une pyramide gaussienne est une pyramide d'images dans laquelle chaque niveau de la pyramide $L_{t,k}$ (sauf $L_{t,0} = L_t$ qui contient l'image originale) est une version filtrée passe-bas et sous-échantillonnée du niveau précédent. Nous considérons le résultat du premier niveau de la pyramide que nous notons

$L_t^{fil} = L_{t,1}$. Pour chaque pixel (i,j) de L_t^{fil} , avec $0 \leq i < \frac{MAX_X}{2}$, $0 \leq j < \frac{MAX_Y}{2}$,

nous avons :

$$L_t^{fil}(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) L_t(2i + m, 2j + n). \quad (3.28)$$

$W(m, n)$ est un noyau générateur séparable défini par:

$$W(m, n) = W(m)W(n), \quad (3.29)$$

où $W(0) = \frac{1}{3}$, $W(1) = W(-1) = \frac{1}{4}$, $W(2) = W(-2) = \frac{1}{12}$.

Au décodeur, l'image L_t^{int} est reconstruite par interpolation de l'image \hat{L}_t^{fil} en utilisant la formule A.3 :

$$L_t^{int}(i, j) = 4 \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) \hat{L}_t^{fil}(i, j) \left(\frac{i-m}{2}, \frac{j-n}{2} \right). \quad (3.30)$$

où $0 \leq i < MAX_X - 1$, $0 \leq j < MAX_Y - 1$, et seuls les termes pour lesquels $\frac{i-m}{2}$ et $\frac{j-n}{2}$ sont entiers sont inclus dans les sommations.

3.2.2 Module de compensation de disparité

Ce module ressemble beaucoup à un module de compensation du mouvement. Il se base sur la méthode décrite dans la section A.3.1.

Pour chaque bloc $B_{L,t}$ de L_t^{fil} , il associe un vecteur représentant le déplacement de ce bloc entre L_t^{fil} et \hat{H}_t^{fil} . Ce vecteur est noté $(disp_x, disp_y)$.

La contrainte épipolaire nous a amené à choisir un espace de recherche rectangulaire de longueur $2 \times \lambda_{disp,x}$ et de hauteur $2 \times \lambda_{disp,y}$ centré autour de la position du bloc courant à coder, mais dans l'image homologue reconstruite.

En supposant les images alignées, la composante horizontale $disp_x$ doit avoir un espace de recherche beaucoup plus étendu que la composante verticale $disp_y$.

De plus, nous nous contentons d'une précision au niveau pixel du vecteur de disparité pour éviter d'avoir à effectuer une interpolation au décodeur. La figure 3.20 illustre le principe générique de notre module de compensation de disparité.

Pour chaque bloc $B_{L,t}(x, y)$ de taille $N \times N$, le meilleur bloc $\hat{B}_{H,t}(x + u, y + v)$ appartenant à la bande de recherche satisfait le critère suivant :

$$EQM = \min_{\text{bande}} \sqrt{\sum_{(k,l)=(0,0)}^{(N-1,N-1)} \frac{1}{N^2} \left[B_{L,t}(x + k, y + l) - \hat{B}_{H,t}(x + disp_x + k, y + disp_y + l) \right]^2}. \quad (3.31)$$

Si cette erreur est inférieure ou égale à un seuil donné Γ_{disp} , la compensation de disparité est satisfaisante et le vecteur $(disp_x, disp_y)$ est transmis au canal. Sinon, le bloc $B_{L,t}$ passe dans le module de codage suivant.

Il faudra prendre garde à la qualité de l'approximation car il ne faudrait surtout pas perdre l'information de relief par une image L_t trop dégradée. Dans le chapitre 4, la qualité de fusion est évaluée par des tests subjectifs.

Par ailleurs, le rôle des deux images doit être interverti régulièrement afin de prévenir le phénomène de fatigue visuelle.

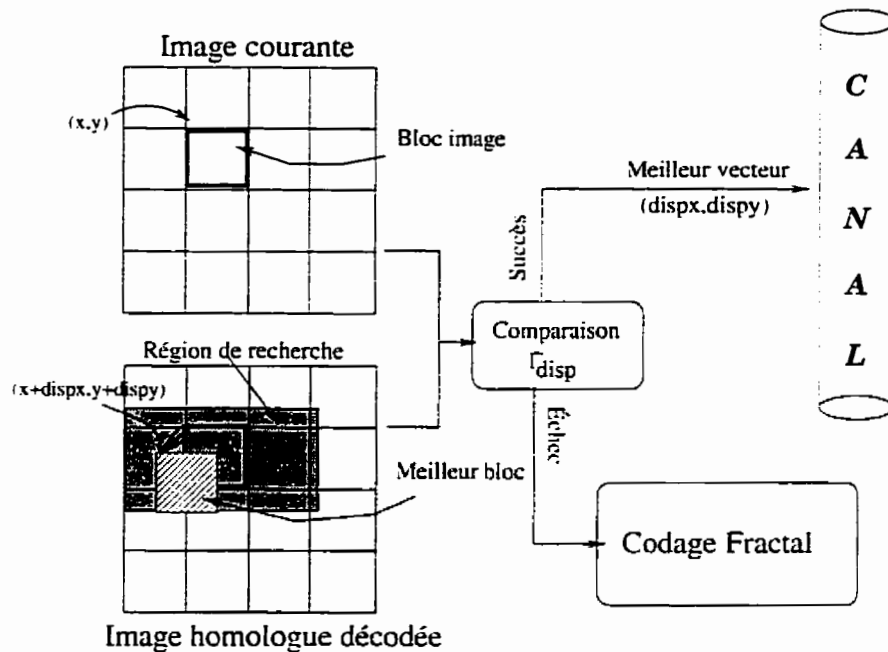


Figure 3.20 : Module de compensation de disparité

3.2.3 Étude de l'influence des seuils et paramètres

L'étude est menée pour 5 itérations successives des séquences à basse résolution des deux séquences stéréoscopiques tests "Papillon" et "Tunnel".

• Influence du seuil de succès Γ_{disp} :

Nous avons étudié l'influence de Γ_{disp} sur la dégradation du point de vue erreur quadratique moyenne de la séquence à basse résolution. Les résultats présentés à la figure 3.21 montrent que, pour les deux séquences, il est plus avantageux de prendre $\Gamma_{disp} = 8$ que des plus petites valeurs. Ces résultats sont similaires à ceux obtenus lors de l'étude de Γ_{mut} . De même, nous devons prendre garde à ne pas prendre Γ_{disp} trop grand car nous rappelons que les artefacts de type "effets de blocs" sont d'autant plus visibles que les blocs sont de grande taille

(plus nombreux lorsque Γ_{disp} est grand).

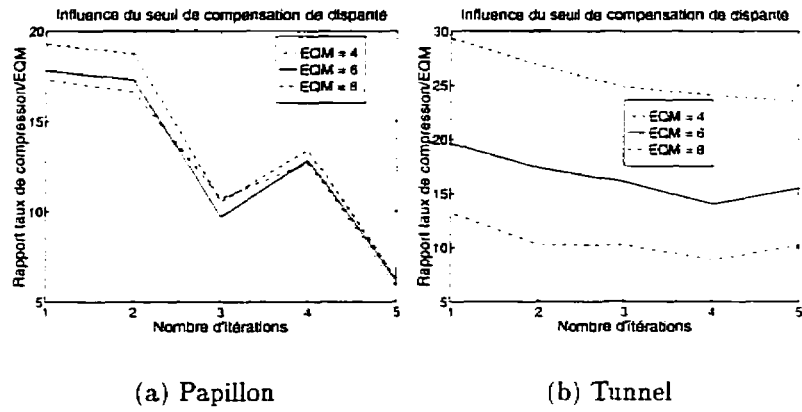


Figure 3.21 : Influence du seuil Γ_{disp} sur le rapport entre le taux de compression et l'erreur quadratique moyenne

- *Influence de la longueur de $\lambda_{disp,x}$:*

La figure 3.22 représentant l'influence de la variation de $\lambda_{disp,x}$ sur le rapport entre le taux de compression et l'erreur quadratique moyenne $\frac{\tau}{EQM}$ montre que $\lambda_{disp,x} = 63$ et $\lambda_{disp,x} = 7$ sont à rejeter pour les deux séquences. Les deux valeurs intermédiaires donnent des résultats équivalents.

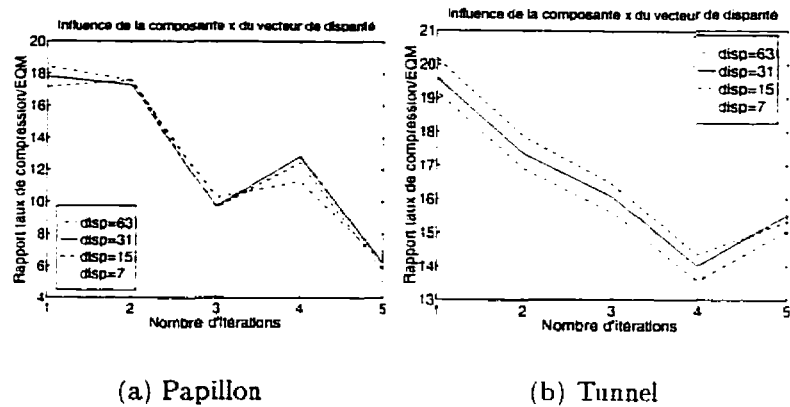


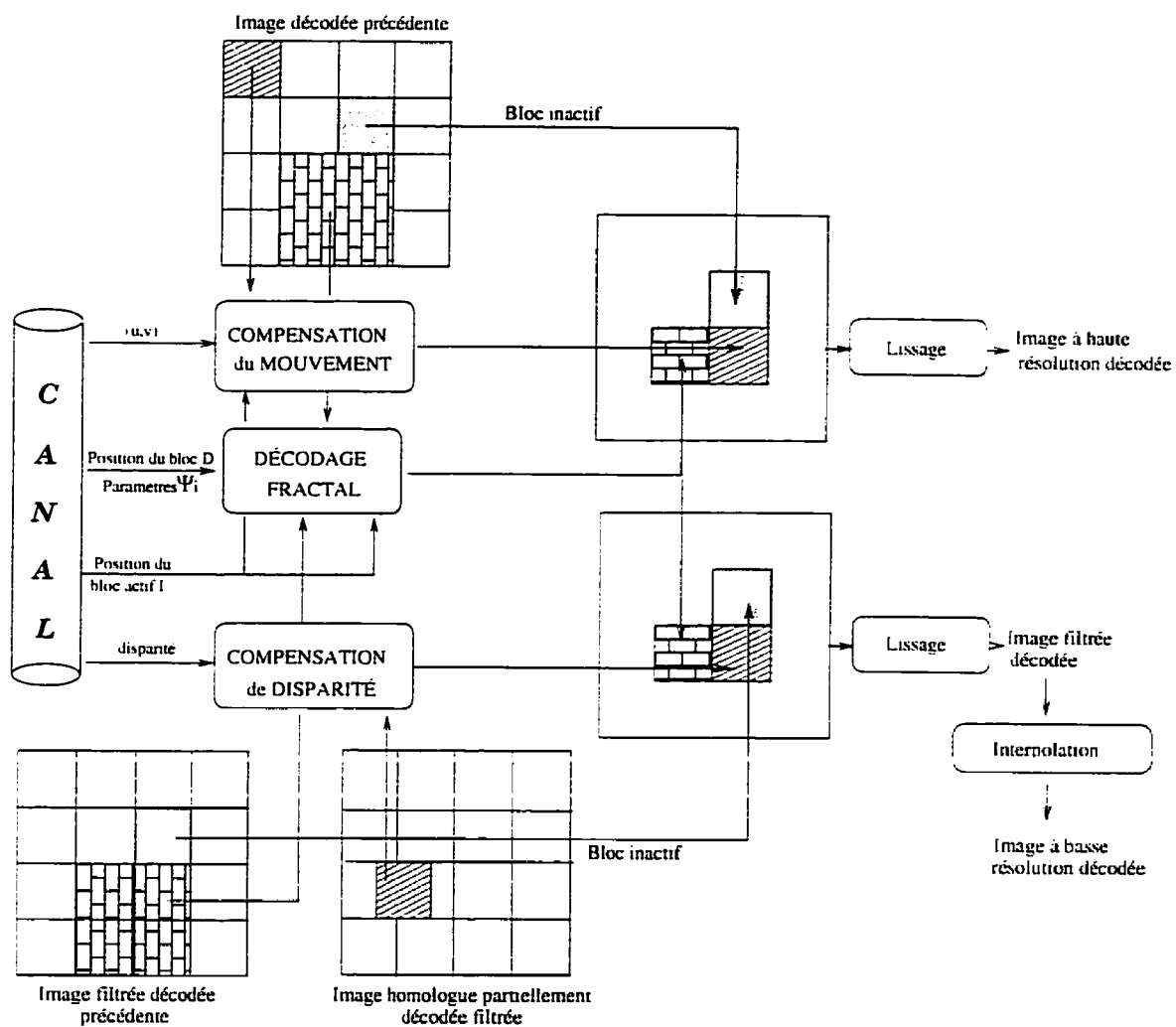
Figure 3.22 : Influence de $\lambda_{disp,x}$ sur le rapport entre le taux de compression et l'erreur quadratique moyenne

3.3 Éléments constitutifs du décodeur

Comme l'illustre la figure 3.23, le décodeur est très simple puisqu'il est constitué d'un module qui oriente de manière adaptative le décodage de chaque bloc. En ce qui concerne le décodage d'une image de la séquence H , le décodeur reçoit d'abord l'information sur la segmentation du bloc actif et donc sa taille et sa position. Il reçoit de plus l'information du type d'approximation du bloc. Si le bloc I a été approximé par le module de compensation du mouvement, le décodeur reçoit l'adresse et la taille du meilleur bloc D de la trame précédente et, à partir de là, positionne le bloc D . Si le bloc I a été approximé par le module de codage fractal, le décodeur reçoit l'adresse du meilleur bloc de la trame précédente ainsi que les paramètres de transformation associés. Dès lors, il peut reconstruire le bloc I .

Par ailleurs, le décodage de l'homologue stéréoscopique de basse-qualité avec compensation du mouvement s'effectue de manière totalement indépendante de l'autre séquence. Par contre, si l'exploitation de la stéréoscopie est intégrée dans un module de compensation de disparité comme c'est illustré à la figure 3.23, le décodage de l'homologue stéréoscopique de basse-qualité s'effectue de manière séquentielle et nécessite un délai équivalent au décodage d'une bande horizontale de blocs. En effet, chaque bloc approximé par compensation de disparité est décodé en utilisant l'information de l'adresse du bloc l'approximant et appartenant à l'image homologue H au même instant. Or, grâce à la contrainte épipolaire, cette adresse est placée sur une même bande horizontale comme expliqué dans la section précédente.

Ce décodeur a été conçu de manière à traiter les blocs dans le domaine spatial et à ne pas être itératif. Toutes les conditions sont donc réunies pour qu'il soit particulièrement rapide.



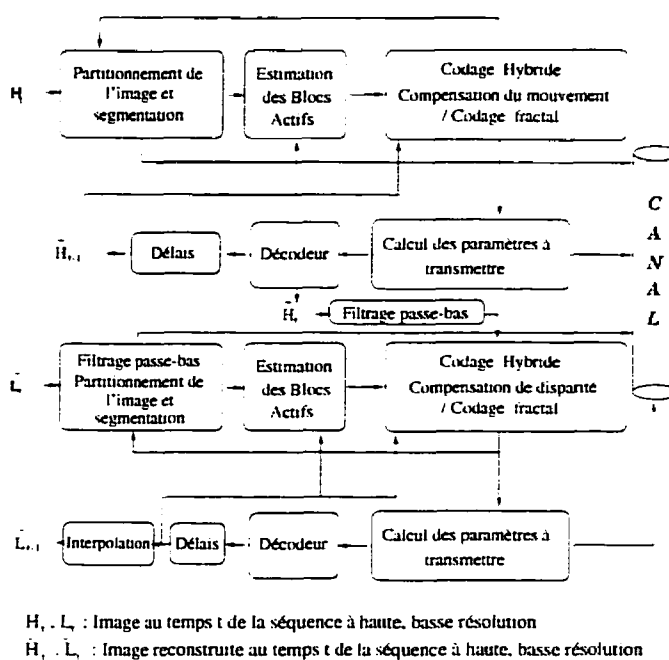
3.4 Bilan

Le tableau 3.2 récapitule l'ensemble des valeurs choisies pour les seuils et paramètres des différents modules constituant le codeur.

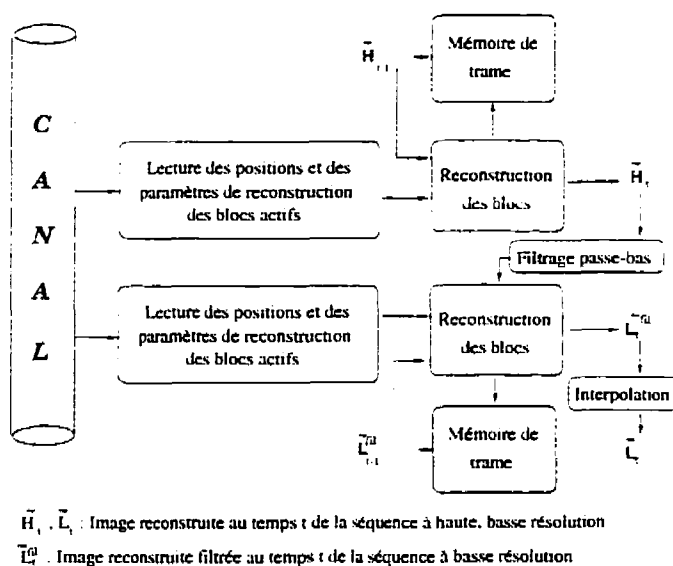
Tableau 3.2 : Ensemble des paramètres du codeur

Paramètre	Valeur
Γ_{mut}	6
λ_{mut} en nombre de bits	5
λ_{frac} en nombre de bits	9
a en nombre de bits	5
b en nombre de bits	6
Δ_{prep}^{max}	5
Γ_{prep}	0.09
Codage multi-niveaux des blocs actifs	oui
Codage multi-niveaux des blocs passant dans le module de compensation du mouvement	oui
Codage multi-niveaux des blocs passant dans le module de codage fractal	non
Γ_{disp}	6
$2 \times \lambda_{disp,x}$ en nombre de bits	6
$2 \times \lambda_{disp,y}$ en nombre de bits	3

La figure 3.24 résume la structure de notre codec.



(a) Codeur



(b) Décodeur

Figure 3.24 : Schéma bloc du codec

3.5 Implantation

L'image initiale I_0 est codée de manière intra-trame par codage fractal seul et en considérant tous les blocs actifs. Par la suite, chaque image est codée ou décodée à partir de l'image précédente reconstruite.

3.5.1 Notations utilisées

Le tableau 3.3 résume l'ensemble des notations utilisées dans cette section pour calculer les taux de compression.

3.5.2 Décodabilité du système

Notre travail consiste à effectuer un codage de source et nous ne nous occupons pas de l'étape suivante de compression et de protection contre les erreurs supplémentaires que constitue le codage de canal. Cependant, nous devons nous préoccuper de la décodabilité du train de bits généré par notre système et de la non-propagation indéfinie d'erreurs de transmission.

3.5.2.1 Différents choix de transmission numérique

Nous appelons *masque* un ensemble de bits d'entête servant à représenter les informations nécessaires pour décoder les blocs partitionnant l'image. Nous avons comparé les résultats du point de vue compression avec trois choix de transmission numériques des paramètres de codage: le mode avec masque de longueur variable (MV), le mode

Tableau 3.3 : Notations utilisées

Symbole	Signification
$MAX_x \times MAX_y$	Taille des images
T	Nombre d'images dans la séquence
Nb	Nombre de blocs ($Nb = \frac{MAX_x MAX_y}{16}$ pour des blocs de 4×4)
N_{ina}^i	Nombre de blocs inactifs de taille $i \times i$
N_{act}	Nombre total de blocs actifs
N_{mvt}^i	Nombre de blocs actifs codés par compensation du mouvement de taille $i \times i$
N_{frac}^i	Nombre de blocs actifs codés par codage fractal de taille $i \times i$
λ_{sync}	Nombre de bits utilisés pour la transmission de la séquence de synchronisation
λ_{pa}	Nombre de bits utilisés pour coder l'entête d'un bloc actif dans le mode PA
λ_m	Nombre de bits utilisés pour coder le masque dans le mode MV $\lambda_m = 3 \times Nb$
λ_{mf}	Nombre de bits utilisés pour coder le masque dans le mode MF
λ_{cod}	Nombre total de bits utilisés pour coder les paramètres de l'ensemble des blocs actifs
λ_{mvt}	Nombre de bits utilisés pour la transmission de la position du meilleur bloc trouvé par compensation du mouvement
λ_{frac}	Nombre de bits utilisés pour la transmission de la position du meilleur bloc domaine
q_a	Nombre de bits utilisés pour la quantification et la protection du facteur d'échelle a
q_b	Nombre de bits utilisés pour la quantification et la protection du facteur de décalage b
q_{iso}	Nombre de bits utilisés pour la transmission et la protection du type d'isométrie

avec masque de longueur fixe (MF) et le mode avec transmission de la position des blocs actifs (PA). Ces systèmes correspondent à des conditions différentes de transmission, mais offrent tous le minimum de protection pour assurer une synchronisation sur chaque image.

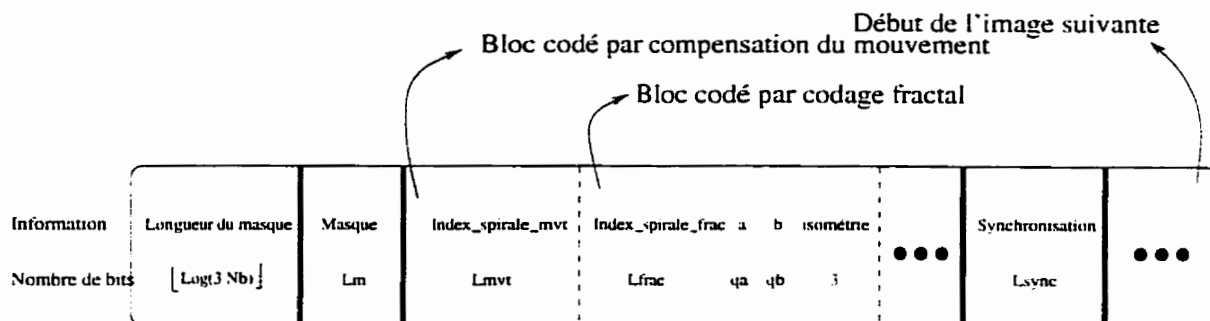
Synchronisation sur chaque image Pour que le décodeur soit capable de retrouver le début de chaque nouvelle image en cas d'erreur et de perte de la synchronisation lors du décodage d'une image, il faut inclure dans le flux binaire une séquence unique (non présente ailleurs dans le train de bits décrivant le codage de l'image) dite de synchronisation. Nous avons choisi cette séquence comme étant une suite de bits à '1'.

Ce choix nécessite de protéger toutes les séquences de '1' consécutives en interdisant aux différents paramètres à transmettre de prendre ces valeurs. Ainsi, certains paramètres se voient alourdis d'un bit supplémentaire de protection :

- λ_{mv} = 5 (il peut prendre 31 valeurs avec la séquence "1 1 1 1 1" protégée):
- λ_{frac} = 9 (il peut prendre 511 valeurs avec la séquence "1 1 1 1 1 1 1 1 1" protégée):
- q_a est fixé à 6 bits (5 bits pour quantifier a et le premier bit fixé à "0"):
- q_b vaut 7 bits (6 bits pour quantifier b et le premier bit fixé à "0").

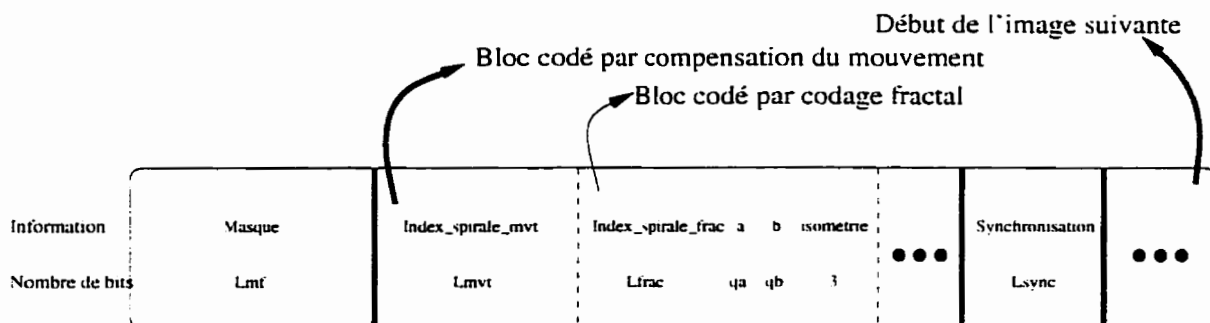
La longueur minimale nécessaire de la séquence de synchronisation doit être calculée en fonction des autres paramètres transmis dans le canal. De plus, nous allons voir qu'elle dépend du type de transmission choisi.

Description des trois modes de transmission Les flux binaires transmis selon chaque mode sont montrés à la figure 3.25.



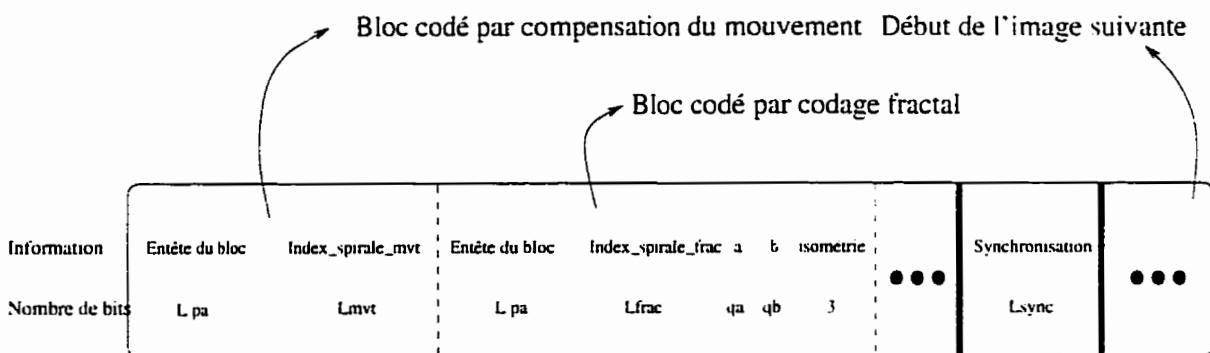
Ensemble des paramètres de codage des blocs actifs

Masque de longueur variable MV



Ensemble des paramètres de codage des blocs actifs

Masque de longueur fixe MF



Position des blocs actifs PA

Figure 3.25 : Flux binaires transmis selon le mode de transmission

Le tableau 3.4 contient, pour chaque catégorie de blocs générés lors de l'étape de codage, le nombre de bits moyens utilisés pour coder leur entête selon le mode de transmission considéré. L'explication des valeurs contenues dans ce tableau est détaillée avec la description de chaque mode.

Tableau 3.4 : Catégories de blocs générés et nombre de bits moyens d'entête avec chaque mode

Type	Mode MV	Mode MF	Mode PA
V_{ina}^{16}	1	3	0
V_{ina}^8	1.5	3	0
V_{ina}^4	2	3	0
V_{mvt}^{16}	2	3	λ_{pa}
V_{mvt}^8	2.5	3	λ_{pa}
V_{mvt}^4	3	3	λ_{pa}
V_{frac}^4	3	3	λ_{pa}

1. Masque de longueur variable (MV) :

Cette méthode est la plus simple à produire car elle correspond directement à la structure en arbre quaternaire produite par le codec. La figure 3.15 illustre l'arbre quaternaire avec le flux binaire qu'il génère.

En cas d'erreur dans le canal, il faut s'assurer que le décodeur sera capable de se recalculer sur le début de la prochaine image en retrouvant la séquence de synchronisation. Or, comme le masque est de longueur variable, il se pourrait qu'au cours de la recherche de la séquence de synchronisation, le décodeur retrouve la même séquence dans le flux binaire composant le masque. Pour empêcher cela,

il faut envoyer au préalable dans le canal la taille du masque.

- Codage de la taille du masque :

La taille du masque est au maximum de $\lfloor \log_2(3 \times Nb) \rfloor$ bits. Ceci correspond au pire cas dans lequel tous les blocs de l'image sont actifs et de taille 4×4 ce qui représente Nb blocs. De plus, l'information "bloc actif de taille 4×4 " est codée sur 3 bits avec l'arbre quaternaire.

- Longueur de la séquence de synchronisation :

La longueur de synchronisation est donnée par l'équation 3.32.

$$\lambda_{sync} = \lambda_{mvt} + \lambda_{frac} - 1 + P \quad (3.32)$$

$\lambda_{mvt} + \lambda_{frac}$ représentent les deux paramètres codés consécutifs les plus longs du flux binaire. Ce sont donc eux qui vont fixer λ_{sync} . Le terme "-1" représente les 2 "0" inclus dans le codage de λ_{mvt} et λ_{frac} et le bit supplémentaire à rajouter pour éviter l'ambiguïté.

P représente le nombre de bits supplémentaires de protection en cas d'erreur dans le canal et de transformation de un bit à '0' en '1'. Dans ce mode, P vaut $\max_{q_{iso}, \lambda_{mvt}}$.

Conformément au tableau 3.4, le nombre de bits comportant le masque

suit l'équation 3.33 :

$$\lambda_m = 1 \times V_{ina}^{16} + 1.5 \times V_{ina}^8 + 2 \times V_{ina}^4 + 2 \times V_{mut}^{16} + 2.5 \times V_{mut}^8 + 3 \times V_{mut}^4 + 3 \times V_{frac}^4. \quad (3.33)$$

- Le nombre total de bits utilisés pour coder les blocs actifs dans une image a pour expression :

$$\lambda_{cod} = \lambda_{mut} \times (V_{mut}^{16} + V_{mut}^8 + V_{mut}^4) + [\lambda_{frac} + q_a + q_b + q_{iso}] \times V_{frac}^4. \quad (3.34)$$

- Le rapport de compression obtenu suit la relation 3.35 :

$$\frac{T \times MAX_x MAX_y \times 8\text{bits}}{[\lambda_{frac} + q_a + q_b + q_{iso}] \times Nb + \sum_{t=1}^T (\lambda_{sync} + \lfloor \log_2(3 \times Nb) \rfloor + \lambda_m(t) + \lambda_{cod}(t))} : \quad (3.35)$$

2. Masque de longueur fixe (MF) :

Ce système fixe la longueur du masque en décrivant chaque bloc 4×4 qui compose l'image avec trois bits correspondant à l'une des sept classes possibles décrites dans le tableau 3.4. Le flux binaire transmis selon ce mode est représenté à la figure 3.25.

- Longueur de la séquence de synchronisation :

Comme dans le cas précédent, elle suit l'expression 3.32.

- Longueur du masque :

Comme expliqué précédemment, $\lambda_{mf} = 3 \times Nb$ bits.

- Le nombre total de bits utilisés pour coder les blocs actifs dans une image suit aussi l'expression 3.34.
- Le rapport de compression obtenu suit la relation 3.36:

$$\frac{T \times MAX_x \cdot MAX_y \times 8\text{bits}}{[\lambda_{frac} + q_a + q_b + q_{iso}]Nb + \sum_{t=1}^T (\lambda_{sync} + 3 \times Nb + \lambda_{cod}(t))} : 1. \quad (3.36)$$

3. *Transmission de la position des blocs actifs :*

Comme nous pouvons le voir à la figure 3.25, dans ce mode de transmission, nous ne transmettons pas de masque: seules les informations relatives aux blocs actifs sont envoyées dans le canal: l'entête de chaque bloc actif et les paramètres de codage associés à ce bloc.

- Longueur de la séquence de synchronisation :

$$\lambda_{sync} = \lfloor \log_2 MAX_x \rfloor + \lfloor \log_2 MAX_y \rfloor - 1 + P. \quad (3.37)$$

Dans ce mode, P vaut $\max_{q_{iso}, \lambda_{mt}}$.

- Le nombre de bits pour coder l'entête d'un bloc actif suit l'équation 3.38:

$$\lambda_{pa} = \lfloor \log_2 (MAX_x) \rfloor + \lfloor \log_2 (MAX_y) \rfloor + 2 + 1. \quad (3.38)$$

Les deux premiers termes représentent le nombre de bits nécessaires pour

coder la position du bloc en x et y. Il faut ensuite ajouter 2 bits pour coder le niveau de segmentation et 1 bit pour coder le type de codage effectué.

- Le nombre total de blocs actifs est :

$$N_{act} = N_{mut}^{16} + N_{mut}^8 + N_{mut}^4 + N_{frac}^4. \quad (3.39)$$

- Le rapport de compression obtenu suit donc la relation 3.40 :

$$\frac{T \times MAX_x MAX_y \times 8bits}{[\lambda_{frac} + q_a + q_b + q_{iso}]Nb + \sum_{t=1}^T (\lambda_{sync} + \lambda_{pa} \times N_{act}(t) + \lambda_{cod}(t))} : 1. \quad (3.40)$$

3.5.2.2 Comparaison des trois modes

Performance du point de vue compression : Selon le nombre et le niveau de segmentation des blocs actifs, chaque mode devance tour à tour les deux autres. Globalement, le premier mode est le plus performant dans le cas général et le troisième mode lorsque le nombre de blocs inactifs de taille 4x4 est très grand.

Résilience aux erreurs de canal : Les trois modes de transmission ont été construits de manière à assurer la synchronisation sur chaque image. En outre, la performance de notre système de codage provient en partie du fait que le codage est mené de manière adaptative. Ceci se traduit dans le train de bits générés par une information binaire de longueur variable pour chaque bloc de l'image.

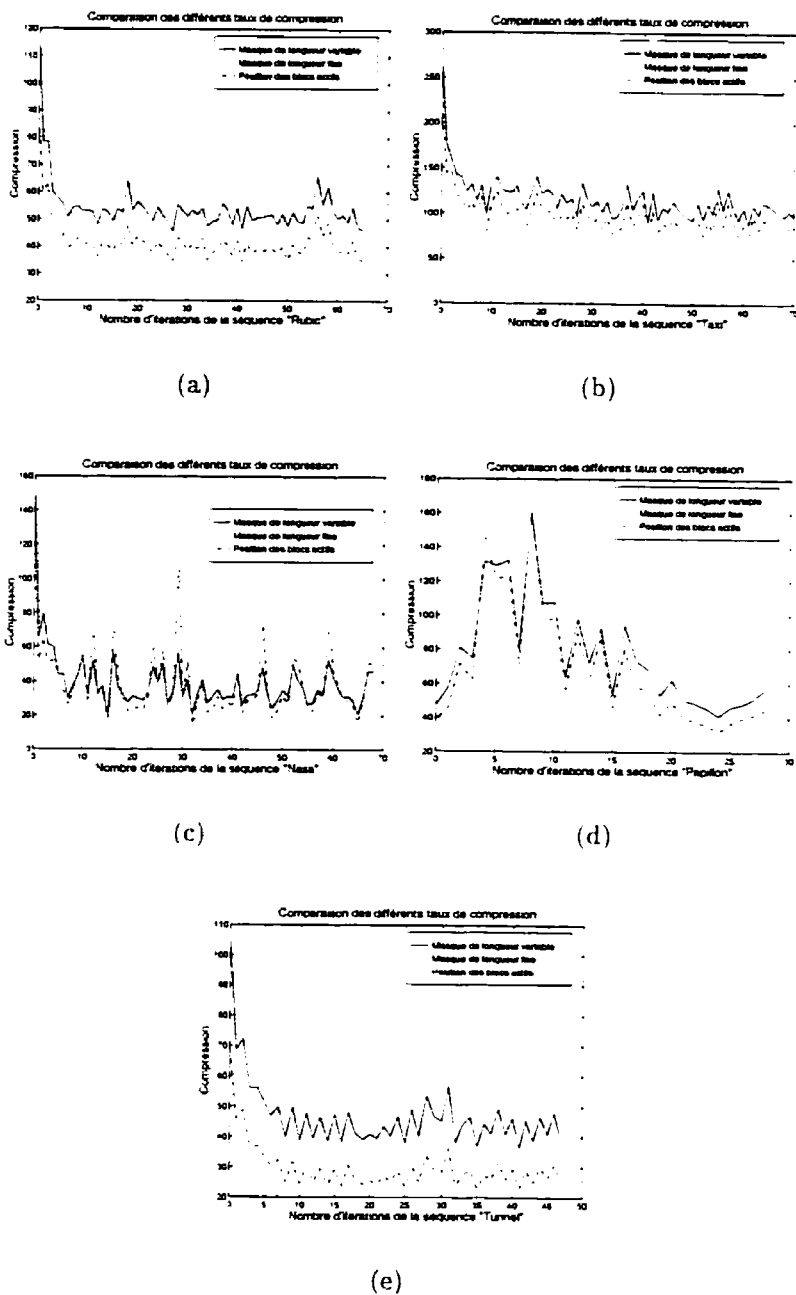


Figure 3.26 : Comparaison des trois modes de transmission

Cette adaptabilité rend notre système plus ou moins sensible selon le mode de transmission choisi aux erreurs de inversion de bits.

Ainsi, le mode *MV* est celui qui donne le meilleur résultat du point de vue compression, mais il est très sensible aux inversions de bits qui se produisent lors de la transmission du masque. En effet, une erreur sur la valeur d'un seul bit peut se propager à tout le reste de l'image. Par exemple, si dans la lecture du masque, une inversion de bits se produit sur le bit d'information de niveau de segmentation d'un bloc, lors du décodage, la taille du bloc décrite par ce bit va être mal interprétée et un décalage de tous les bits qui suivent vont propager cette erreur à toute l'image. Le mode *PA* présente la même sensibilité que le mode *MV* pour ce qui est de la sensibilité au niveau de segmentation et au type de codage. Par contre, le mode *MF* considère tous les blocs de la même taille; il n'est donc pas sensible au niveau de segmentation. Par contre, comme les deux autres modes, il est très sensible aux erreurs se produisant sur un bit relatif au type de codage effectuée car ce bit conduit à la lecture d'un nombre différent de bits dans le train de bits et donc à un décalage s'il a été erroné.

Des erreurs de type insertion de bits sont fatales et se répercutent à tout le reste de l'image pour les trois modes. Il faudrait donc développer un système de protection des bits relatifs au niveau de segmentation et au type de codage effectué en utilisant, par exemple, des systèmes de contrôles de parité par des techniques de codes correcteurs d'erreurs.

Chapitre 4

Résultats expérimentaux

Cette section contient les résultats de différentes expériences monoculaires et stéréoscopiques menées sur les séquences tests présentées dans l'annexe B. Cette dernière contient également la dernière image reconstruite de chaque séquence test pour chacune de ces expériences.

4.1 Codage de séquences d'images monoculaires

4.1.1 Méthodologie

Deux types de tests de qualité ont été effectués:

1. des tests dits **objectifs** se basant sur une mesure de qualité absolue. Nous avons choisi d'utiliser le rapport signal sur bruit maximum noté $PSNR$, défini dans l'équation 4.1, comme paramètre de qualité des images reconstruites au

récepteur. Le rapport $PSNR$ s'exprime en décibels (dB). A titre indicatif, un rapport $PSNR = 30dB$ correspond à une erreur $EQM = 8$.

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{EQM^2}\right). \quad (4.1)$$

L'autre critère de qualité utilisé est le taux de compression global τ obtenu pour chaque image de la séquence. Enfin nous avons considéré le produit $\tau \times PSNR$ comme critère conjoint des deux critères absolus. Plus ce rapport est élevé, meilleure est la qualité de l'image.

2. des tests dits **subjectifs**.

Il est évident que le rapport signal sur bruit maximum est un critère qui ne tient pas compte du système visuel humain. En effet, nous pouvons avoir des images ayant une erreur globale plus faible que d'autres mais dont l'erreur est plus dérangeante pour un observateur humain.

Nous avons considéré deux sortes de tests subjectifs :

- (a) Le choix forcé qui consiste à montrer deux séquences à des observateurs en leur demandant de dire laquelle ils préfèrent. L'ordre dans lequel sont montrées les séquences est aléatoire et les sujets sont contraints à un choix.
- (b) Une échelle d'évaluation de la dégradation visuelle en 7 points qui est représentée sur le tableau 4.1. Nous demandons à l'observateur de donner un chiffre entre 1 et 7 de comparaison de qualité entre la séquence qu'il

est en train de visualiser et la séquence originale. Cette dernière lui a été présentée au préalable et lui est montrée de nouveau dès qu'il en formule le souhait.

Tableau 4.1 : Echelle de visibilité des dégradations

<i>Numérotation</i>	<i>Signification</i>
1	non perceptibles
2	juste perceptibles
3	légères mais définitivement perceptibles
4	non dérangeantes
5	quelque peu dérangeantes
6	définitivement dérangeantes
7	extrêmement dérangeantes

Les séquences ont été observées par trois sujets sur une station Silicon Graphics de manière répétitive et selon un ordre variable.

4.1.2 Justification d'un mode de codage hybride

Pour évaluer la méthode hybride combinant la compensation de mouvement et le codage fractal, nous avons comparé la qualité des images codées avec compensation du mouvement seulement, avec un codage fractal seulement et avec la méthode hybride. Selon l'approche utilisée, le taux de compression était réévalué de manière à ne considérer que les paramètres utiles à transmettre au récepteur. De plus, les trois modes étaient menés en utilisant les mêmes conditions d'ajustement de seuil et de paramètres.

Dans aucun des trois modes l'erreur de prédiction n'était transmise. Cette contrainte handicape le mode de codage par compensation du mouvement qui repose sur des techniques différentielles et est très sensible aux accumulations d'erreur. Cependant, pour que les trois modes puissent être comparés, il faut se placer dans les mêmes conditions de codage.

La dernière image reconstruite de chaque séquence selon les trois modes de codage est représentée sur les figures B.2 et B.3.

4.1.2.1 Tests objectifs

Les trois modes—foncièrement différents dans leur approche—ont des taux de compression parfois très différents d'un mode à l'autre ce qui rend les comparaisons parfois difficiles. Aussi, nous avons considéré comme critère supplémentaire le produit $\tau \times PSNR$ car il cumule les deux paramètres de rapport signal sur bruit maximum et de taux de compression avec une importance égale.

Les figures 4.1, 4.2 et 4.3 montrent le rapport $PSNR$, τ et le produit $\tau \times PSNR$ respectifs obtenus pour les 5 séquences tests. Le taux de compression considéré est celui obtenu sans utiliser aucune protection en cas d'erreur de canal (voir section 3.5.2.1).

Nous pouvons observer que, pour les 5 séquences, le mode "compensation du mouvement pur" génère une accumulation d'erreur inexistant dans les autres modes. En outre, pour les séquences "NASA", "Papillon" et "Tunnel", la figure 4.1 montre que

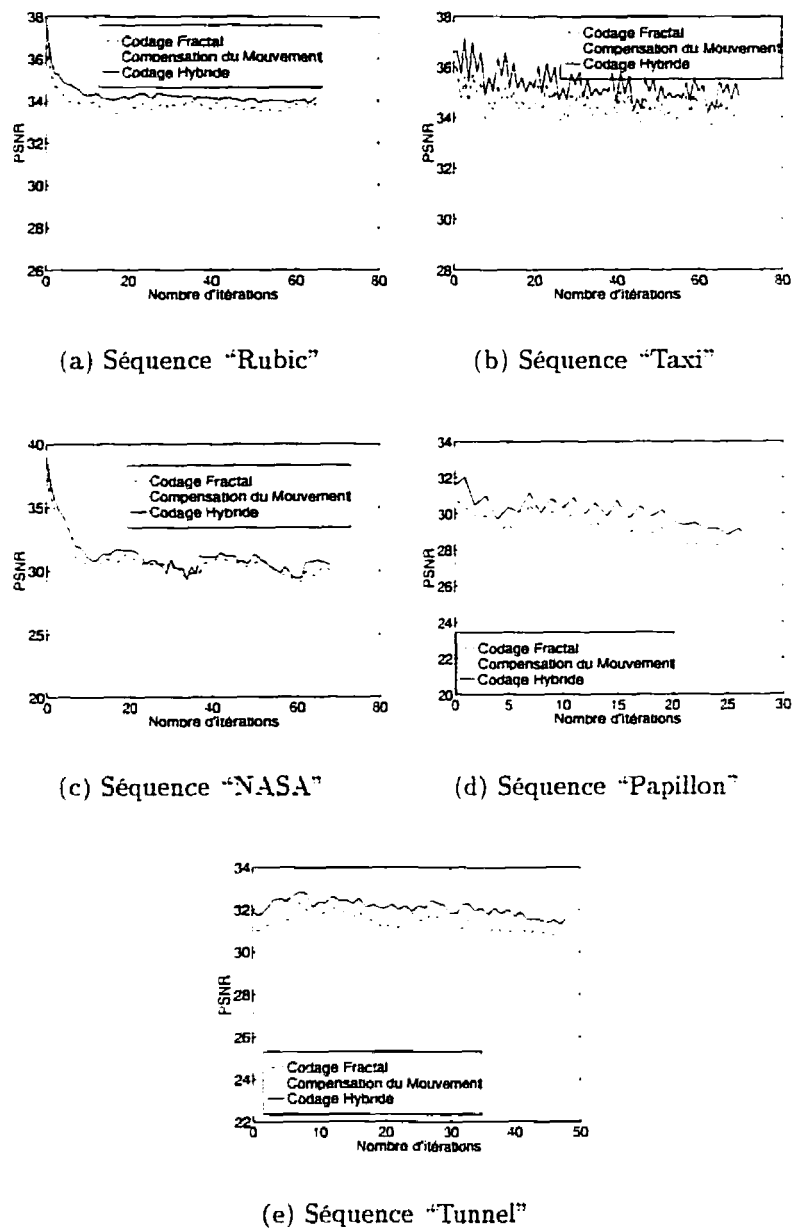


Figure 4.1 : Rapport $PSNR$ obtenu selon les trois modes de codage

la qualité de l'image reconstruite se dégrade dramatiquement au fil des itérations.

Ceci s'explique par le fait que ces trois séquences comportent des mouvements non

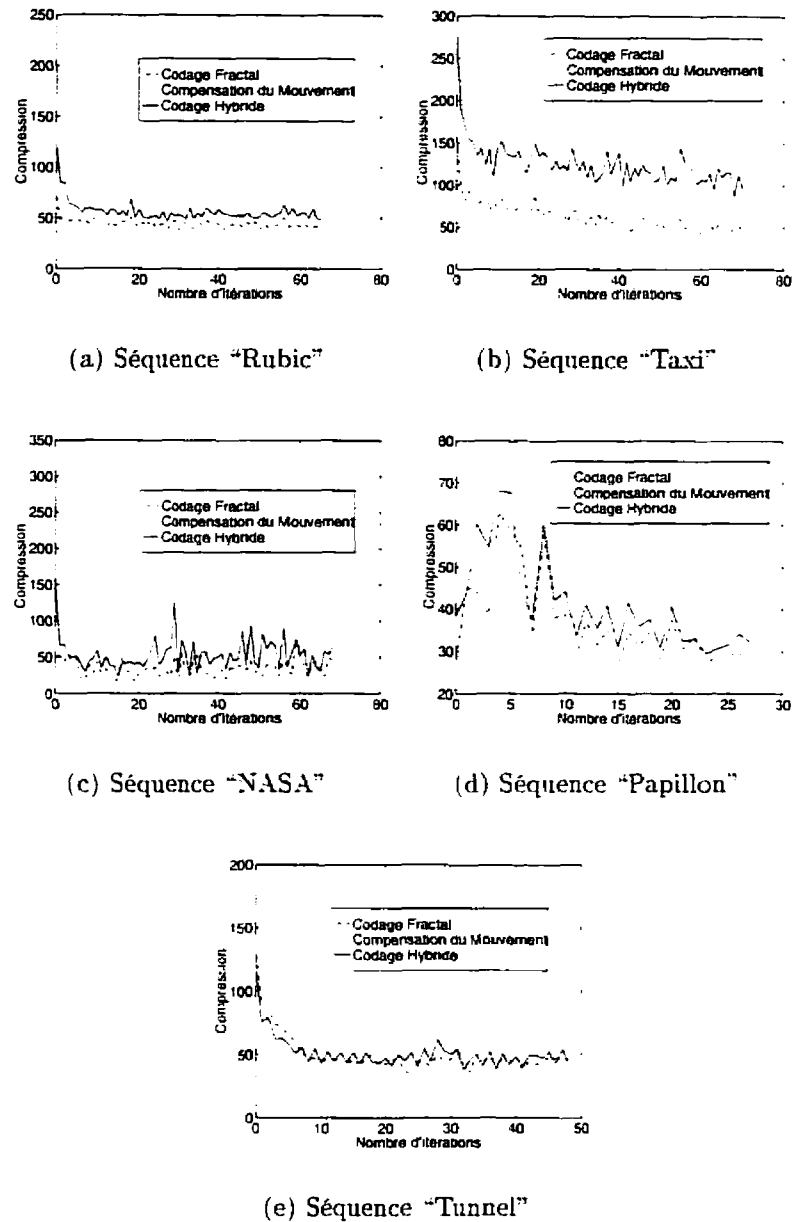


Figure 4.2 : Taux de compression τ obtenu selon les trois modes de codage translationnels impossible à coder par ce mode. Ceci montre un des avantages d'utiliser le codage fractal, qui ne requiert pas le codage de l'erreur de prédiction. Par contre, le mode "codage fractal pur" possède un pouvoir de compression relative-

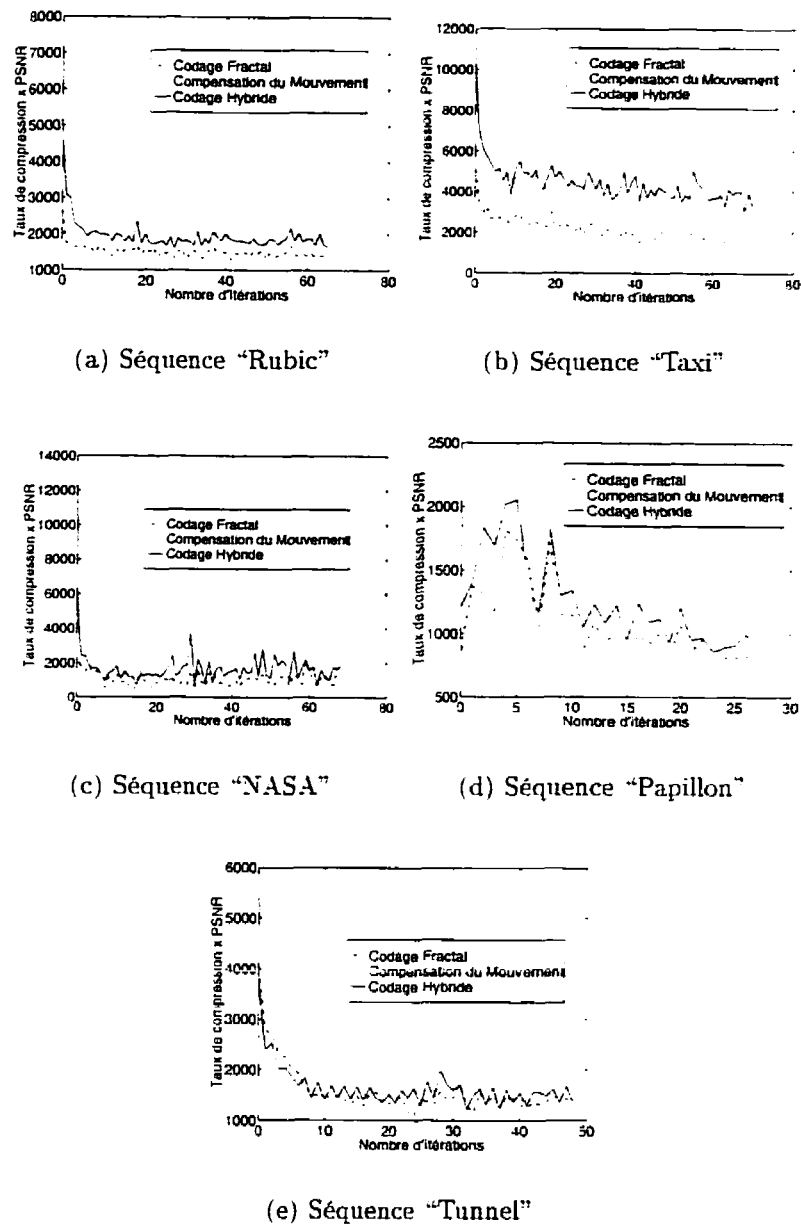


Figure 4.3 : Produit $\tau \times PSNR$ obtenu selon les trois modes de codage

ment pauvre et en-deça des autres modes. Ceci est du au fait que le codage fractal n'est mené que pour des blocs actifs de taille 4×4 . Or, il requiert un nombre de paramètres à transmettre constant quelque soit la taille du bloc codé.

Nous pouvons également remarquer que les courbes correspondant aux modes hybride et fractal se suivent globalement de très près avec toujours une légère supériorité pour le mode hybride. Pour la séquence "Taxi", nous pouvons observer à la figure 4.3 l'avantage substantiel d'ajouter un module de compensation du mouvement dans le mode hybride. Le taux de compression augmente beaucoup car cette séquence comporte des mouvements de type translationnels bien codés par compensation du mouvement.

Une série de tests subjectifs vont permettre de compléter ces remarques.

4.1.2.2 Tests subjectifs

Les tests d'évaluation de la visibilité de la dégradation en 7 points ont été effectués sur trois sujets pour les 5 séquences test. Chacun des trois modes ainsi que la séquence originale ont été présentés 2 fois à chaque observateur dans un ordre aléatoire. Les tableaux 4.2, 4.3, 4.4, 4.5 et 4.6 récapitulent les résultats obtenus. Ceux-ci confirment les commentaires de la section 4.1.2.1. Pour toutes les séquences, le mode hybride donne de meilleurs résultats subjectifs que les deux autres modes. Selon les séquences, notre méthode de codage est plus ou moins performante par rapport à l'originale. Le mode par compensation du mouvement pur est toujours évalué comme générateur de dégradations au moins dérangeantes pour l'observateur à cause du phénomène d'accumulation d'erreurs qu'il exhibe ou de son incapacité à représenter des mouvements complexes. Pour la séquence "Taxi", le mode fractal pur donne les pires résultats

des trois modes car les artefacts présents (effets de blocs dans les zones d'intensité uniforme) sont très dérangeants visuellement. En outre, cette séquence possède une zone d'intérêt (en mouvement) très restreinte ce qui fait que le point de fixation de l'observateur repère instantanément l'artefact. De plus, comme cette séquence ne comporte que des mouvements de translation, le mode de codage par compensation du mouvement performe mieux comparativement aux autres séquences. Pour les trois séquences "NASA", "Papillon" et "Tunnel", comme prévu par les résultats de la section 4.1.2.1, le mode de codage par compensation du mouvement est très mauvais. Les tests objectifs montrent donc que, quelle que soit la séquence considérée, il s'est avéré profitable aussi bien du point de vue compression que qualité de combiner un codage par compensation du mouvement et codage fractal. En effet, l'utilisation du codage fractal permet de ne pas avoir à coder l'erreur de prédiction car l'accumulation d'erreur est limitée. En outre, il permet de coder efficacement des zones de la scène animées d'un mouvement complexe. La compensation du mouvement permet d'améliorer la compression en particulier lorsque le mouvement contenu dans la scène suit le modèle translationnel considéré. Enfin, un aspect qui n'a pas été étudié de manière quantitative, mais qui est important: l'ajout d'un module de compensation du mouvement permet d'accélérer le temps de calcul au codeur puisque certains blocs n'auront pas besoin de passer dans le module de codage fractal.

Outre la comparaison entre les différents modes, il est également intéressant de remarquer la qualité de codage relative du mode de codage hybride par rapport à la

Tableau 4.2 : Résultats subjectifs internes sur la séquence "Rubic"

<i>Séquence</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
Originale	1	1	1	1
Codage fractal seul	2.5	2	2	2.16
Compensation du mouvement seule	3.5	5	4	4.16
Codage hybride	1.5	1.5	2	1.66

Tableau 4.3 : Résultats subjectifs internes sur la séquence "Taxi"

<i>Séquence</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
Originale	1	1	1	1
Codage fractal seul	3	5	6	4.66
Compensation du mouvement seule	4	4	5	4.33
Codage hybride	3	4.5	4.5	4

séquence originale. Selon les séquences, notre méthode présente des dégradations au pire quelque peu dérangeantes ce qui est satisfaisant. Dans la section 4.1.3, nous allons commenter plus précisément les artefacts générés par notre méthode de codage hybride comparée avec celle d'une méthode de codage par compensation du mouvement avec transmission de l'erreur de prédiction: *MPEG - 1*.

Tableau 4.4 : Résultats subjectifs internes sur la séquence "NASA"

<i>Séquence</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
Originale	1	1	1	1
Codage fractal seul	2.5	2	4.5	3
Compensation du mouvement seule	6	4	6	5.33
Codage hybride	2	2.5	3	2.5

Tableau 4.5 : Résultats subjectifs internes sur la séquence "Papillon"

<i>Séquence</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
Originale	1	1	1	1
Codage fractal seul	3.5	3.5	5	4
Compensation du mouvement seule	6.5	7	7	6.83
Codage hybride	2.5	4	4.5	3.66

Tableau 4.6 : Résultats subjectifs internes sur la séquence "Tunnel"

<i>Séquence</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
Originale	1	1	1	1
Codage fractal seul	5	6	6	5.66
Compensation du mouvement seule	7	7	7	7
Codage hybride	4	5	5	4.66

4.1.3 Comparaison de notre système avec *MPEG - 1*

Pour valider l'intérêt de notre méthode pour des travaux futurs, nous avons comparé les résultats obtenus selon les deux modes d'évaluation décrits précédemment (voir 4.1.1) entre notre système et une version software de *MPEG - 1*.

La dernière image reconstruite de chaque séquence en utilisant notre système et deux versions de *MPEG - 1* pour deux taux de compression différents notés *MPEG - 11* (pour le taux de compression le plus faible) et *MPEG - 12* est représentée sur les figures B.4 et B.5.

4.1.3.1 Tests objectifs

Les figures 4.4 montrent les $PSNR$ obtenus pour des taux de compression moyens pour les 5 séquences tests avec notre méthode (en utilisant le mode hybride). $MPEG-1$ et $MPEG-12$. Les valeurs de compression moyennes données sur les figures 4.4 avec notre méthode sont celles obtenues en utilisant un masque de longueur variable (voir section 3.5.2.1).

Les résultats varient selon les séquences. Notre méthode donne un bien meilleur $PSNR$ avec une meilleure compression pour la séquence "Taxi" car celle-ci présente peu de blocs actifs et donc, le potentiel de compression, grâce au module d'estimation des blocs actifs, est excellent.

Pour la séquence "Rubic", les deux méthodes se valent avec cependant un léger avantage pour notre méthode car il présente davantage de stabilité de qualité le long de la séquence que $MPEG-1$.

La séquence "NASA" est une séquence difficile à coder car toute l'image bouge selon un mouvement de convergence ou de divergence. Ceci explique que notre méthode donne une qualité de l'ordre de celle de $MPEG-1$, mais pour un taux de compression légèrement moins bon (car beaucoup de blocs sont actifs).

Pour la séquence "Papillon", les deux méthodes se valent encore. En outre, une accumulation d'erreur est observée pour les deux méthodes qui peut être compensée dans notre méthode en envoyant une image de mise à jour. Ceci s'explique par la complexité de cette séquence qui comprend un fond fortement texturé et animé d'un

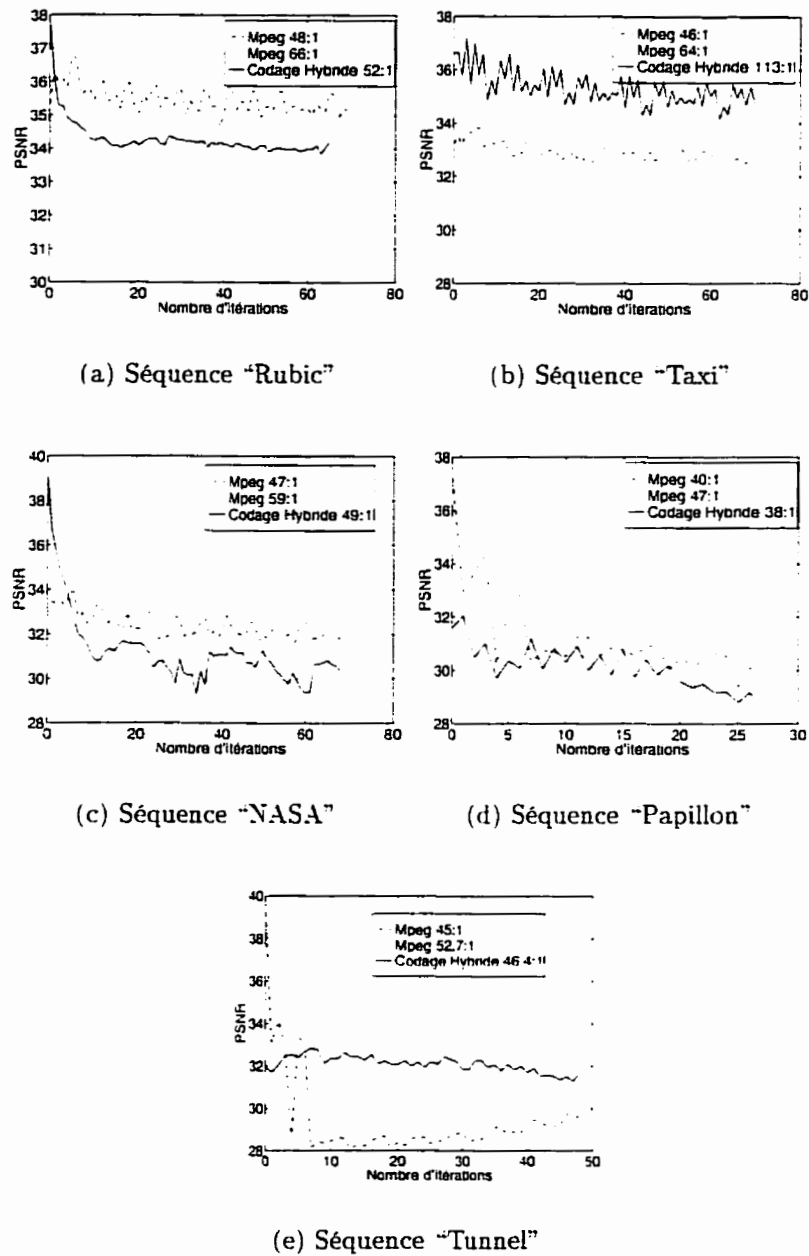


Figure 4.4 : Rapport $PSNR$ obtenu avec notre système et $MPEG - 1$

léger mouvement. L'envoi de cette image de mise à jour ne devrait pas nuire au taux de compression global car le nombre de blocs actifs chutera par la même occasion.

Enfin, notre méthode donne encore de meilleurs résultats pour la séquence "Tunnel". Il est à noter que la version utilisée de *MPEG-1* a eu beaucoup de mal à coder cette séquence à des taux de compression moyens de l'ordre de 50:1. La qualité résultante en a souffert. Ceci s'explique par le fait que cette séquence comprend des mouvements complexes de rotation, de divergence et des phénomènes d'occlusion ce qui la rend difficile à coder avec une bonne compression par *MPEG-1*.

4.1.3.2 Tests subjectifs

Les tests subjectifs de type choix forcé ont été effectués sur trois sujets. La liste des tests effectués sur chacun de ces sujets est présentée au tableau 4.7.

Tableau 4.7 : Liste des tests subjectifs effectués sur chacun des sujets

Première séquence	Deuxième séquence	Nombre de fois
Notre système	<i>MPEG-1</i> 1	4
Notre système	<i>MPEG-1</i> 2	4
Total		8

Les 8 tests précédents ont été présenté de façon aléatoire aux sujets. De plus, l'ordre de présentation de la première et la deuxième séquence a été aussi aléatoire. Les tableaux 4.8, 4.9, 4.10, 4.11 et 4.12 récapitulent les résultats obtenus. Nous pouvons voir que les trois séquences "Rubic", "Taxi" et "NASA" exhibent le même genre de résultats. Globalement, notre système est toujours préféré à *MPEG-1* 2 et est équivalent à *MPEG-1* 1. Ceci montre la supériorité du point de vue qualité visuelle objective de notre système sur *MPEG-1* pour un taux

de compression moyen égal voire un peu supérieur (avec notre système).

Il est à noter cependant que le choix forcé entre notre système et *MPEG - 1* n'est pas le fruit d'une décision aléatoire, mais d'une préférence marquée variable selon les sujets. En effet, nous pouvons remarquer que les scores ne résultent jamais du hasard car ils n'ont pas la valeur 2/4. Ceci signifie que les deux séquences exhibent chacune des artefacts plus ou moins dérangeants selon les sujets.

Ainsi, pour la séquence "Rubic", le troisième sujet a préféré *MPEG - 1* car notre système générerait un scintillement des intensités lumineuses autour des carrés du cube. Pour la même séquence, le deuxième sujet a davantage été dérangé par un artefact affectant le bas du socle, généré par *MPEG - 1*. Enfin, le premier sujet a trouvé la séquence codée avec notre système plus floue.

La séquence "NASA" codée par *MPEG - 1* génère des effets de blocs plus visibles qu'avec notre méthode.

Pour la séquence "Taxi", les sujets 2 et 3 rejettent notre système à cause d'un scintillement des intensités des blocs adjacents sur le fond uniforme d'une voiture blanche. Une manière d'éviter cet artefact serait d'effectuer un lissage des intensités des blocs dans les zones uniformes de l'image.

Contrairement à ce que donnent les tests objectifs, les séquences "Papillon" et "Tunnel" obtiennent des résultats très mauvais. Dans le cas de la séquence "Papillon", ceci est dû principalement à un artefact très gênant consistant en des carrés noirs apparaissant en bordure de l'aile du papillon. Ces derniers surviennent sur les dernières

itérations de la séquence. Nous pouvons d'ailleurs observer la chute substantielle du critère *PSNR* à la figure 4.4. Ces dernières images sont déterminantes et provoquent le choix quasi systématique de *MPEG - 1*. Un autre artefact dérangeant généré par notre système de codage est un mouvement de bruit du fond texturé (fleurs) de la séquence. Ceux-ci sont dus à la classification de ces régions comme inactives ce qui provoque des sauts d'intensité. Ces phénomènes pourraient être évités par l'envoi d'une image de mise à jour au cours de la séquence.

Finalement, pour la séquence "Tunnel", le même type d'artefact que pour la séquence "Taxi" a été noté par les trois sujets. Les effets de blocs existant dans la zone uniforme du wagon sont très dérangeants car ils sont placés dans la zone d'intérêt de l'image et sont permanents. Le système *MPEG - 1 2* génère des artefacts bien plus grossiers sous la forme de blocs noirs épars dans l'image. Pourtant, les observateurs tolèrent davantage ces erreurs à cause de leur localisation en-dehors de la région de fixation.

Tableau 4.8 : Résultats subjectifs externes sur la séquence "Rubic"

<i>MPEG - 1</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
48 : 1	0/4	4/4	0/4	4/12
66 : 1	4/4	4/4	4/4	12/12

Tableau 4.9 : Résultats subjectifs externes sur la séquence "Taxi"

<i>MPEG - 1</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
46 : 1	4/4	1/4	0/4	5/12
64 : 1	4/4	4/4	3/4	11/12

Tableau 4.10 : Résultats subjectifs externes sur la séquence "NASA"

<i>MPEG - 1</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
47 : 1	3/4	3/4	0/4	6/12
59 : 1	4/4	4/4	4/4	12/12

4.2 Codage de séquences d'images stéréoscopiques

4.2.1 Dispositif d'affichage

Nous utilisons une station SGI avec un dispositif de gestion des fenêtres d'affichage stéréos. L'observateur porte des lunettes polarisantes Crystal Eyes lui permettant de recevoir sur chaque oeil l'image qui lui est destinée.

4.2.2 Comparaison de la performance de diverses variantes

Nous comparons les résultats obtenus en utilisant trois variantes possibles de codage de la séquence à basse résolution :

Tableau 4.11 : Résultats subjectifs externes sur la séquence "Papillon"

<i>MPEG - 1</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
40 : 1	0/4	0/4	0/4	0/12
47 : 1	3/4	0/4	0/4	3/12

Tableau 4.12 : Résultats subjectifs externes sur la séquence "Tunnel"

<i>MPEG - 1</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
45 : 1	0/4	0/4	0/4	0/12
52.7 : 1	3/4	1/4	0/4	4/12

1. La variante 1 effectue un filtrage par pyramide gaussienne de la séquence à basse résolution et utilise un codage par compensation du mouvement.
2. De même, la variante 2 effectue un filtrage par pyramide gaussienne de la séquence à basse résolution, mais exploite, de plus, explicitement la géométrie stéréoscopique en utilisant un module de compensation de disparité.
3. Finalement, la variante 3 ne filtre pas la séquence à basse résolution. Elle utilise un codage par compensation de disparité et exploite la théorie de la neutralisation en tolérant des distorsions beaucoup plus importantes pour approximer les blocs au fil des modules de codage.

La dernière image reconstruite de chaque séquence en utilisant les trois variantes de codage est représentée à la figure B.6. A titre de comparaison, nous présentons également à l'observateur la séquence stéréoscopique originale non codée.

Pour évaluer la qualité des séquences stéréoscopiques reconstruites, nous avons utilisé le critère d'évaluation de la dégradation en 7 points qui ressemble à celui utilisé par Schertz (Schertz 1991). Chacun des 4 types de codage sont présentés 2 fois à l'observateur dans un ordre aléatoire. Les résultats obtenus sont illustrés par les tableaux 4.13 et 4.14. Il est à noter que la séquence à haute résolution utilisée est celle du mode

hybride considérée dans les tests monoculaires précédents.

Les résultats obtenus pour les deux séquences sont très encourageants par rapport à nos prévisions. Nous pouvons voir que, malgré un taux de compression plus élevé de la séquence à basse résolution, la version filtrée avec la variante 1 ou 2 est toujours préférée à la variante 3. En effet, pour améliorer le taux de compression dans la variante 3, il a fallu tolérer des distorsions plus élevées et les artefacts générés diminuent la qualité perçue. En outre, dans le cas des variantes 1 et 2, les trois sujets ont déclaré percevoir la séquence fusionnée avec la qualité de la séquence à haute résolution.

Le sujet 3 a déclaré ressentir une fatigue visuelle lorsqu'il observait trop longtemps une séquence codée en utilisant une des deux premières variantes.

Il est à remarquer que les deux séquences stéréoscopiques tests utilisées étaient celles qui donnaient les moins bons résultats dans les séries de tests monoculaires. En fait, lors de leur évaluation, les trois sujets notaient la qualité des séquences visualisées selon les deux critères combinés de qualité de fusion et de visibilité des artefacts.

La fusion était toujours de très bonne qualité pour les trois variantes de codage. Les artefacts de codage apparaissant sous la forme de grosses taches étaient toujours plus visibles dans la variante 3. Pour l'observateur 3, ces taches apparaissaient dans un plan plus rapproché que le plan de convergence et ne gênaient pas la fusion, mais la qualité perçue.

Les variantes 1 et 2 donnent globalement la même qualité de résultats ce qui laisse le choix à l'utilisateur selon l'application. La variante 1 conduit à un taux de compres-

sion un peu moins bon que la variante 2 mais permet le codage/décodage des deux séquences en parallèle ce que ne permet pas la variante 2.

Les résultats obtenus tant pour les séquences monoculaires que stéréoscopiques sont globalement satisfaisants. Les artefacts générés par notre méthode de codage suggèrent quelques améliorations possibles et précautions à considérer que nous décrivons dans la conclusion.

Tableau 4.13 : Résultats subjectifs stéréoscopiques sur la séquence “Papillon”

<i>Séquence</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
Originale	1	1	1	1
Variante 1 à 265 : 1	1	2	3	2
Variante 2 à 256 : 1	1	2	3	2
Variante 3 à 85.5 : 1	3	5	5	4.33

Tableau 4.14 : Résultats subjectifs stéréoscopiques sur la séquence “Tunnel”

<i>Séquence</i>	<i>Sujet 1</i>	<i>Sujet 2</i>	<i>Sujet 3</i>	<i>Moyenne</i>
Originale	1	1	1	1
Variante 1 à 260 : 1	2	3	3	2.66
Variante 2 à 435 : 1	2	2	4	2.66
Variante 3 à 274.5 : 1	2	5	4	3.66

Conclusion

Nous avons développé un système de compression de séquences d'images stéréoscopiques présentant les caractéristiques et originalités qui suivent.

La principale originalité de notre méthode réside dans la structure hybride de notre système. La combinaison du codage par compensation du mouvement et du codage fractal a déjà été considérée par d'autres auteurs (Ali, Clarckson et Papadopoulos 1992, Hürtgen et Büttgen 1993, Andonova et Popovic 1994, de Faria et Ghanbari 1995) mais dans des applications différentes de la nôtre. Notre optique étant de développer un système offrant une qualité visuelle satisfaisante pour des débits de transmission moyens et des délais de codage faibles, il nous a fallu trouver un moyen d'accélérer l'étape de codage. Pour cela, nous avons choisi d'utiliser un modèle translationnel simple pour le module de compensation du mouvement. Par ailleurs, le processus de codage fractal mené dans l'espace des fréquences a permis d'accélérer le codeur par la construction originale d'une représentation invariante contrôlée par une étude de compatibilité de signes. Cette dernière permet d'assurer que le meilleur bloc calculé approxime de manière satisfaisante le bloc courant dans le domaine spatial.

En outre, bien qu'introduisant des calculs supplémentaires de comparaison de signes, cette étude n'augmente pas le temps de calcul au codeur car elle permet de réduire le nombre de blocs domaines candidats à considérer ce qui économise des calculs inutiles. En outre, elle rend le codage adaptatif car chaque bloc est codé de manière particulière en considérant seulement ses termes fréquentiels prépondérants.

L'usage combiné de la compensation du mouvement et du codage fractal permet de satisfaire nos critères de qualité et de compression. Notre méthode présente les avantages du codage fractal de ne pas dépendre de la complexité du mouvement dans la scène et de ne pas nécessiter la transmission de l'erreur de prédiction. Par contre, le fait de coder certains blocs par compensation du mouvement améliore la compression mais supprime la propriété fractale d'indépendance en résolution de l'image décodée. La structure de notre système le rend peu sensible au type de mouvement dans la scène et adaptatif à différentes applications. En effet, notre codeur s'adapte localement à tous les types de redondance pouvant exister dans une séquence d'images stéréoscopiques: spatiale, temporelle, fractale et stéréoscopique. En outre, la qualité de l'approximation atteinte est de plus en plus fine (avec un taux de compression de plus en plus bas) au fil des modules (estimation des blocs actifs, compensation du mouvement ou de disparité, codage fractal et segmentation). Cette propriété permet d'obtenir différentes qualités d'image en ajustant les paramètres de distorsion tolérables. Nous avons la possibilité d'effectuer un contrôle du débit binaire pour des applications à très bas débit.

L'autre contribution de notre travail se situe dans la stratégie de codage stéréoscopique. Une fois compilés les nombreux travaux exploitant la théorie de la neutralisation dans l'optique du codage de paires d'images stéréoscopiques (Dinstein, Guy, Rabany, Tselgov et Henik 1989, Dinstein, Kim, Tselgov et Henik 1991, Perkins 1992, Zhu et al. 1991), nous avons adapté et développé leurs méthodes pour le codage de séquences d'images stéréoscopiques. Une des deux séquences est filtrée passe-bas et sous-échantillonnée au moyen d'une pyramide gaussienne ce qui diminue la taille de chaque image de la séquence de moitié selon chaque dimension. Ensuite, deux variantes de codage ont été envisagées. La version filtrée de la séquence peut être codée selon le même principe que la séquence à haute résolution, ou bien en remplaçant le module de compensation du mouvement par un module de compensation de disparité. La première option permet le codage/décodage des deux séquences en parallèle. Par contre, la deuxième variante utilise explicitement¹ la géométrie stéréoscopique existant entre les séquences homologues. Ceci permet d'améliorer la compression mais rend la reconstruction de la séquence à faible résolution dépendante séquentiellement de l'autre séquence.

Par ailleurs, notre méthode de codage asymétrique préserve une séquence à haute résolution. Cette dernière peut donc être regardée de manière monoculaire.

Des tests objectifs (basés sur le rapport signal sur bruit maximum) et subjectifs (utilisant des critères de qualité perceptuels) ont permis d'évaluer la performance de notre

¹dans les deux variantes, la stéréoscopie est exploitée de manière implicite par le fait de filtrer une des deux séquences

système. Une première série de comparaisons internes ont prouvé la prévalence d'un système hybride de codage sur un mode de compression pur par compensation du mouvement ou codage fractal seul. Une deuxième série de tests a servi à comparer notre système avec le standard *MPEG - 1*. Ils ont mis en évidence le fait que notre système est généralement meilleur que *MPEG - 1* du point de vue *PSNR* pour des taux de compression égaux. Par contre, les tests subjectifs montrent que les deux méthodes génèrent des artefacts différents et que la préférence de l'une ou l'autre méthode dépend des observateurs.

Enfin, une dernière série de tests subjectifs effectuée sur des séquences tests stéréoscopiques ont validé la théorie de la neutralisation. Ces tests ont permis de vérifier que lorsque le cerveau est soumis à une séquence d'images stéréoscopiques dans laquelle une des deux séquences a été filtrée, il génère une séquence tri-dimensionnelle de qualité égale à celle de la séquence non filtrée. En outre, nous avons considéré deux variantes de codage effectuant, soit de la compensation du mouvement, soit de la compensation de disparité, qui génèrent globalement la même qualité d'images. Ces deux variantes sont toujours préférables à une méthode de codage sans filtrage de la séquence à basse résolution.

Des améliorations et ouvertures souhaitables pourraient être considérées dans le but d'augmenter la performance de ce système.

Une image de mise à jour codée de manière intra-trame ² par codage fractal pour-

²c'est-à-dire utilisant l'information disponible dans la même image

rait être envoyée afin de pallier aux accumulations d'erreurs de codage et aux erreurs de canal. La décision d'envoyer une image de mise à jour serait basée sur l'erreur quadratique moyenne globale de l'image reconstruite. En outre, un lissage des intensités des blocs dans les zones uniformes permettrait de réduire davantage les effets de blocs visibles avec notre méthode (Nakajima, Hori et Kanoh 1994, Bedford, Dekking et Keane 1994).

La segmentation en arbre quaternaire devrait être basée sur un critère entropique et non seulement sur l'erreur quadratique moyenne pour tenir compte de la diminution de compression à des niveaux de segmentation plus petits (Dufaux et Moscheni 1995, Lee 1995). En outre, cette segmentation pourrait être de type rectangulaire ou de type polygonale avec fusion des blocs ressemblants qui seraient codés de la même manière (Thomas et Deravi 1995).

Pour mieux contrôler le débit binaire, nous pourrions utiliser des seuils de distorsion adaptatifs pour chaque module de codage.

Pour réduire les effets de blocs, nous pourrions considérer un partitionnement de l'image avec superposition des blocs. Les intensités des pixels localisés dans une zone commune à plusieurs blocs seraient calculés comme la somme pondérée de la contribution de l'intensité fournie par le même pixel de chacun des blocs (Young et Kingsbury 1993, Reusens 1994).

En ce qui concerne le module de compensation du mouvement, la recherche du meilleur bloc pourrait être accélérée par une méthode logarithmique (Bhaskaran et

Konstantinides 1995) et en tenant compte des résultats obtenus pour les blocs voisins déjà codés. Ceci est validé par le fait que des blocs voisins ont des chances d'être soumis à des mouvements proches et donc d'avoir un vecteur de translation quasi-similaire.

Enfin, pour ce qui est du module de compensation de disparité, il serait intéressant de quantifier la disparité dépendamment de la distance. En effet, la disparité peut être quantifiée de façon plus grossière lorsque sa valeur est grande car le système visuel humain est moins sensible à des objets placées loin du plan de convergence (Schertz 1991).

Bibliographie

- ALI, M., CLARCKSON, T. et PAPADOPOULOS, C. (1992). The use of fractal theory in a video compression scheme, dans J. Storer, éditeur. *IEEE Data Compression Conference*, pp. 259–268.
- ANDONOVA, S. et POPOVIC, D. (1994). Video coding using fractal-based image compression, dans *1994 IEEE International Conference on Systems, Man, and Cybernetics: Humans, Information, and Technology*, Vol. 1, pp. 343–348.
- ARDITI, A. (1986), *Handbook of Perception and Performance: Sensory Processes and Perception*, Vol. 1, Wiley, New York, chapter Binocular Vision, pp. 23, 1–23, 41.
- ARYA, S., MOUNT, D. M., NETANYAHU, N. S., SILVERMAN, R. et WU, A. (1994). An optimal algorithm for approximate nearest neighbor search, dans *Proc 5th Annual ACM-SIAM Symposium on Discrete Algorithms (1994)*, pp. 573–582.
- BALDINE-BRUNEL, P. et HAYES, M. H. (1994), Fractal-based compression of motion video sequences, dans *IEEE International Conference on Image Processing (ICIP '94)*, Atlanta, Georgia, pp. 755–759.

- BANI-EQBAL, B. (1995), Speeding up fractal image compression, dans *Proceedings of SPIE: Still-Image Compression*, Vol. 2418, pp. 67-74.
- BARNSLEY, M. F. (1988), *Fractals Everywhere*. Academic Press. San Diego.
- BARTHEL, K., NOLL, P. et VOYÉ, T. (1993), Improved fractal image coding, dans *Proceedings from Picture Coding Symposium (PCS)*, section 1. 5.
- BARTHEL, K., SCHUTTEMEYER, J., VOYÉ, T. et NOLL, P. (1994), A new image coding technique unifying fractal and transform coding, dans *IEEE International Conference on Image Processing (ICIP '94)*, Austin, Texas, pp. 112-116.
- BARTHEL, K. U. et VOYÉ, T. (1995), Three-dimensional fractal video coding, dans *IEEE International Conference on Image Processing (ICIP '95)*, Atlanta, Georgia, pp. 260-263.
- BARTHEL, K. et VOYÉ, T. (1993), Adaptive fractal image coding in the frequency domain, dans *Proceedings of the International Workshop on Image Processing. Theory, Methodology, Systems and Applications*, Budapest, Hongrie, Juin 1993.
- BARTOLINI, F., CAPPELLINI, V., MECOCCHI, A. et VAGHEGGI, R. (1994), A segmentation-based motion-compensated scheme for low-rate video coding, dans *Proceedings of the IEEE International Conference on Image Processing*, Vol. 2, pp. 457-461.

- BEAUMONT, J. M. (1991), Image data compression using fractal techniques, *British Telecom Technology Journal* 9(4), pp. 93–109.
- BEDFORD, T., DEKKING, F. M. et KEANE, M. S. (1994), Fractal coding of monochrome images, *SignalProcessing* 6, pp. 405–419.
- BHASKARAN, V. et KONSTANTINIDES, K. (1995), *Image and Video Compression Standards: Algorithms and Architectures*. Kluwer Academic Publishers.
- BLAKE, R. (1989), A neural theory of binocular rivalry, *Psychological Review* 96(1), pp. 145–167.
- BOGDAN, A. (1994a), Multiscale fractal image coding and the two-scale difference equation. rapport technique CU/CTR/TR 358-94-05. Columbia University.
- BOGDAN, A. (1994b), Multiscale (inter/intra frame) fractal video coding, dans *Proc. IEEE International Conference on Image Processing. ICIP '94*, Austin, TX.
- BOGDAN, A. et MEADOWS, H. E. (1992), Kohonen neural network for image coding based on Iteration Transformation Theory, *SPIE Neural and Stochastic Methods in Image and Signal Processing* 1766, pp. 425–436.
- BONNAUD, L. et LABIT, C. (1994), Etude d'algorithmes de suivi temporel de segmentation basée mouvement pour la compression de séquences d'images. rapport technique 2253, INRIA.

- BRACEWELL, R. N., CHANG, K. Y., WANG, A. K. et WANG, Y. H. (1993). Affine theory for 2-dimensional Fourier transform, *Electronics Letters* **29**(3), pp. 304.
- BÜRGER, W. et BHANU, B. (1992). *Qualitative Motion Understanding*. Kluwer academic publishers.
- CHAN, M., YU, Y. et CONSTANTINIDES, A. (1990). Variable size block matching motion compensation with applications to video coding, dans *IEEE Proceedings. Part 1, Communications, Speech and Vision*, Vol. 137, pp. 205–212.
- DAVOINE, F., SVENSSON, J. et CHASSERY, J.-M. (1995). A mixed triangular and quadrilateral partition for fractal image coding, dans *IEEE International Conference on Image Processing 1995 (ICIP)*, Vol. 3, pp. 284–287.
- DESKNUYDT, A., DESMET, S., EYCKEN, L. V. et OOSTERLINCK, A. (1994). A human visual system-based block classification algorithm for image sequence coders, dans *Proceedings of the SPIE: Visual Communication and Image Processing (Part 2)*, Vol. 2308, pp. 1401–1410.
- DIAB, Z. et COHEN, P. (1996). Compression vidéo par compensation du mouvement – revue bibliographique, rapport technique GRPR-RT-9615, École Polytechnique de Montréal.
- DINSTEIN, I., GUY, G., RABANY, J., TSELGOV, J. et HENIK, A. (1989). On the compression of stereo images: Preliminary results, *Signal Processing* **17**(4), pp. 373–382.

- DINSTEIN, I., KIM, M., TSELGOV, J. et HENIK, A. (1989), Compression of stereo images and the evaluation of its effects on 3d perception, dans *Proceedings of the SPIE: Applications of Digital Processing XII*, Vol. 1153, pp. 522-530.
- DINSTEIN, I., KIM, M., TSELGOV, J. et HENIK, A. (1991), Compression of stereo images using subsampling and transform coding, *Optical Engineering* 30(9), pp. 1359-1364.
- DUDBRIDGE, F. (1994), Fast image coding by a hierarchical fractal construction, *submitted* pp. 1-6.
- DUFAUX, F. et MOSCHENI, F. (1995), Motion estimation techniques for digital tv: A review and a new contribution, *Proceedings of the IEEE* 83(6), pp. 858-876.
- DUFAUX, F., MOSCHENI, F. et LIPPMAN, A. (1995), Spatio-temporal segmentation based on motion and static segmentation, dans *Proceedings of the IEEE International Conference on Image Processing*, Vol. 1, pp. 306-309.
- FARIA, S. D. et GHANBARI, M. (1994), Variable block size fractal video coding with spatial transform motion compensation, dans *Proceedings from Picture Coding Symposium (PCS)*, pp. 437-439.
- FARIA, S. D. et GHANBARI, M. (1995), Hybrid fractal/DCT coding of video, dans *5th International Conference on Image Processing and its Applications (IEE)*, pp. 321-325.

- FEIG, E., PETERSON, H. et RATNAKAR, V. (1995), Image compression using spatial prediction, dans *Proceedings of the ICASSP*, pp. 2339–2342.
- FISHER, Y. (1995), *Fractal Image Compression: Theory and Application*. Springer-Verlag.
- FISHER, Y., SHEN, T. P. et ROGOVIN, D. (1994a), A comparison of fractal methods with DCT (JPEG) and wavelets (EPIC), dans *SPIE Proceedings. Neural and Stochastic Methods in Image and Signal Processing III*, Vol. 2304-16, San Diego, CA, Juillet 28-29, 1994.
- FISHER, Y., SHEN, T. P. et ROGOVIN, D. (1994b), Fractal (self-VQ) encoding of video sequences, dans *Proceedings of the SPIE: Visual Communications and Image Processing*, Vol. 2304-16, Chicago, IL, Septembre 28-29, 1994.
- FRIEDMAN, J. H., BENTLEY, J. L. et FINKEL, R. A. (1977), An algorithm for finding best matches in logarithmic expected time, *ACM Transactions on Mathematical Software* **3**(3), pp. 209–226.
- FROBERT, C. et COHEN, P. (1996a), Compression d'images et de séquences vidéo par codage fractal—revue bibliographique, rapport technique GRPR-RT-9613, École Polytechnique de Montréal.
- FROBERT, C. et COHEN, P. (1996b), Exploitation de la théorie de la neutralisation pour le codage d'images stéréoscopiques, rapport technique GRPR-RT-9614, École Polytechnique de Montréal.

- FUH, C. et MARAGOS, P. (1991), Motion displacement estimation using an affine model for image matching, *Optical Engineering* **30**(7), pp. 881–887.
- GALL, D. L. (1991), Mpeg: A video compression standard for multimedia applications, *Communications of the ACM* **34**(4), pp. 47–58.
- GHARAVI-ALKHANSARI, M. et HUANG, T. (1994a), Fractal-based techniques for a generalised image coding method, dans *IEEE International Conference on Image Processing*, pp. 122–126.
- GHARAVI-ALKHANSARI, M. et HUANG, T. (1994b), Generalised image coding using fractal-based methods, dans *Proceedings of the International Picture Coding Symposium*, pp. 440–443.
- GOTTING, D., IBENTHAL, A. et GRIGAT, R. (1995), Fractal image coding and magnification using invariant features, dans *Conf. Proc. NATO ASI Fractal Image Encoding and Analysis*, Trondheim, juillet 1995.
- HAMZAOUI, R. (1995), Codebook clustering by self-organizing maps for fractal image compression, *Fractals* pp. 30–41.
- HANG, H., SCHILLING, D. et PURI, A. (1987), Interframe coding with variable block-size motion compensation, dans *IEEE Global Telecommunications Conference*, pp. 65–69.

- HUANG, S., CHEN, J., WANG, J., HSIEH, K. et HSIEH, H. (1994). Classified variable block size motion estimation algorithm for image sequence coding, dans *Proceedings of the IEEE International Conference on Image Processing*, Vol. 3, pp. 736-740.
- HUANG, Y. et ZHUANG, X. (1995). An adaptively refined block matching algorithm for motion compensated video coding, *IEEE Transactions on Circuits and Systems for Video Technology* 5(1), pp. 56-59.
- HURD, L., GUSTAVUS, M. et BARNSLEY, M. (1992). Fractal video compression, dans *Proceedings of the 37th IEEE Computer Society International Conference (COMPCON)*, Vol. 37, pp. 41-42.
- HÜRTGEN, B. et BÜTTGEN, P. (1993). Fractal approach to low rate video coding, dans *Proceedings of the SPIE: Visual Communication and Image Processing*, Vol. 2094, Boston, MA, USA, pp. 120-131.
- HÜRTGEN, B. et HAIN, T. (1994). On the convergence of fractal transforms, dans *Conference Proceedings of ICASSP*, Vol. 5, pp. 561-564.
- HÜRTGEN, B. et STILLER, C. (1993). Fast hierarchical codebook search for fractal coding of still images, dans *EOS/SPIE Visual Communication and PACS for Medical Applications*, Berlin, Germany, pp. 397-408.
- ISO (1990). Coding of moving pictures and associated audio. *Committee Draft of Standard ISO11172: ISO/MPEG 90/176*.

- JACOBS, B., FISHER, Y. et BOSS, R. (1992), Image compression: A study of the iterated transform method, *Signal Processing* **29**, pp. 251-263.
- JACQUIN, A. E. (1992), Image coding based on a fractal theory of iterated contractive image transformations, *IEEE Transactions on Image Processing* **1**(1), pp. 18-30.
- JULESZ, B. (1971), *Foundations of Cyclopean Perception*, The University of Chicago Press.
- KAPLAN, S. (1992), Compression standards and alternative methods for video telecom, *IEEE Aerospace and Electronics Systems Magazine* **7**, pp. 27-30.
- KAWAMATA, M., NAGAHISA, M. et HIGUCHI, T. (1994), Multi-resolution tree search for iterated transformation theory-based coding, dans *IEEE International Conference on Image Processing*, pp. 137-141.
- KIM, C.-S., KIM, R.-C. et LEE, S.-U. (1995), Novel fractal image compression method with non-iterative decoder, dans *IEEE International Conference on Image Processing (ICIP '95)*, pp. 268-271.
- KIM, I. et PARK, R. (1994), Image coding based on fractal approximation and vector quantization, dans *IEEE International Conference on Image Processing*, pp. 132-136.

- KISELYOV, O. et FISHER, P. (1994), Self-similarity of multiresolutional image video decomposition: Smart expansion as compression of still and moving pictures. dans J. Storer, éditeur, *IEEE Data Compression Conference*, pp. 331-340.
- KOMINEK, J. (1995), Algorithm for fast fractal image compression. dans *Proceedings from IST/SPIE 1995 Symposium on Electronic Imaging: Science and Technology, Digital Video Compression: Algorithms and Technologies*, Vol. 2419, pp. 296-305.
- LABONTÉ, F. et LAGANIÈRE, R. (1996), Stereokineopsis: A survey, rapport technique GRPR-RT-9603, École Polytechnique de Montréal.
- LAZAR, M. S. et BRUTON, L. T. (1994), Fractal block coding of digital video. *IEEE Transactions on Circuits and Systems for Video Technology* 4(3), pp. 297-308.
- LEE, J. (1995), Optimal quadtree for variable block size motion estimation. dans *Proceedings of the IEEE International Conference on Image Processing*, Vol. 3, pp. 480-483.
- LEONARDI, R. et CHEN, H. (1994), Tree based motion compensated video coding. dans *Proceedings of the IEEE International Conference on Image Processing*, Vol. 2, pp. 438-442.
- LEPSOY, S. et OIEN, G. E. (1994), Fractal-based image coding with fast decoder convergence. *Signal Processing* 40, pp. 105-117.

- LEPSOY, S., ØIEN, G. E. et RAMSTAD, T. (1993). Attractor image compression with a fast non-iterative decoding algorithm. dans *Proceedings of the ICASSP*. Vol. 5. pp. 337-340.
- LEWIS, A. S. et KNOWLES, G. (1992). Image compression using the 2-d wavelet transform. *IEEE Transactions on Image Processing* 1(2). pp. 244-250.
- LI, H. et FORCHHEIMER, R. (1995). A transformed block-based motion compensation technique. *IEEE Transactions on Communications* 43(4). pp. 1673-1676.
- LI, H., NOVAK, M. et FORCHHEIMER, R. (1993). Fractal-based image sequence compression scheme. *Optical Engineering* 32(7). pp. 1588-1595.
- LIN et VENETSANOPOULOS, H. (1994). An image coding scheme using block prediction of the pyramid subband decomposition. dans *IEEE International Conference on Image Processing (ICIP '94)*.
- LIN et VENETSANOPOULOS, H. (1995). A pyramid algorithm for fast fractal image compression. dans *IEEE International Conference on Image Processing (ICIP '95)*. pp. 596-600.
- LIU, J., MARLOW, S. et MURPHY, N. (1993). Multi-level fractal block coding in video compression. dans *Proceedings of Amsterdam Conference 1993 DSP-The enabling technology for communication (ERA)*.

- MONRO, D. (1993). A hybrid fractal transform. dans *IEEE Proceedings of ICASSP: Image and Multi-Dimensional Signal Processing*, Vol. V. pp. 169–172.
- MONRO, D. M. et NICHOLLS, J. A. (1994). Real time fractal video for personal communications. *Fractals* 2(6), pp. 39–42.
- MONRO, D. M. et WOOLLEY, S. J. (1994). Fractal image compression without searching, dans *Proceedings of ICASSP 1994*, Vol. V. pp. 557–560.
- MOON, Y., SON, K., KIM, H. et KIM, J. (1995). Fractal image compression using human visual system. dans *Proceedings of the SPIE: Still-Image Compression*, Vol. 2418. pp. 60–66.
- NAKAJIMA, Y., HORI, H. et KANO, T. (1994). A pel adaptive reduction of coding artifacts for mpeg video signals, dans *Proceedings of the IEEE International Conference on Image Processing*, Vol. 2. pp. 928–932.
- NAKAYA, Y. et HARASHIMA, H. (1994). Motion compensation based on spatial transformations, *IEEE Transactions on Circuits and Systems for Video Technology* 4(3), pp. 339–356.
- OHTA, M. et NOGAKI, S. (1993). Hybrid picture coding with wavelet transform and overlapped motion-compensated interframe prediction coding, *IEEE Transaction on Signal Processing* 41(12), pp. 3416–3424.

OIEN, G. E., BAHARAV, Z., LEPSØY, S., KARNIN, E. et MALAH, D. (1994).

A new improved collage theorem with applications to multiresolution fractal image coding, dans *International Conference on Acoustics, Speech, and Signal Processing*, submitted.

PAPADOPOULOS, C. et CLARKSON, T. (1993), Use of 2nd order geometric transformations for motion compensation in interframe image data compression, dans *Proceedings of the IEEE Global Telecommunications Conference*, Vol. 2, pp. 1309–1313.

PARDÀS, M., SALEMBIER, P. et GONZALEZ, B. (1994), Motion and region overlapping estimation for segmentation-based video coding, dans *Proceedings of the IEEE International Conference on Image Processing*, Vol. 2, pp. 428–432.

PERKINS, M. (1992), Data compression of stereopairs, *IEEE Transactions on Communications* **40**(4), pp. 684–696.

RATNAKAR, V., FEIG, E. et TIWARI, P. (1994), Fractal based hybrid compression schemes, dans *Proceedings of the SPIE: Visual Communication and Image Processing*, Vol. 2308, pp. 448–454.

REUSENS, E. (1993), Sequence coding based on the fractal theory of iterated transformations systems, dans *Proceedings of the SPIE: Visual Communications and Image Processing*, Vol. 2094, Boston, USA, pp. 132–140.

- REUSENS, E. (1994), Overlapped adaptive partitioning for image coding based on the theory of iterated functions systems, dans *International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, Adelaide, Australia, pp. 569–572.
- SALEMBIER, P., TORRES, L., MEYER, F. et GU, C. (1995), Region-based video coding using mathematical morphology, *Proceedings of the IEEE* **83**(6), pp. 843–857.
- SAUPE, D. (1994a), Accelerating fractal image compression by multi-dimensional nearest neighbor search, dans J. Storer, éditeur, *IEEE Data Compression Conference (DCC'95)*, pp. 222–231.
- SAUPE, D. (1994b), Breaking the time complexity of fractal image compression, rapport technique 53, Institut für Informatik, University of Freiburg.
- SAUPE, D. (1995), Fractal image compression via nearest neighbor search, dans Y. Fisher, éditeur, *Conf. Proc. NATO ASI Fractal Image Encoding and Analysis*, Trondheim, juillet 1995.
- SAUPE, D. et HAMAZAOUI, R. (1994a), Complexity reduction methods for fractal image compression, dans J. M. Blackledge, Oxford University Press, éditeur, *IMA Conf. Proc. on Image Processing; Mathematical methods and Applications*.
- SAUPE, D. et HAMAZAOUI, R. (1994b), A guided tour of the fractal compression literature, rapport technique, Course Notes – Course 13: New Directions for Fractal Modeling in Computer Graphics, University of Freiburg.

- SCHERTZ, A. (1991), Source coding of stereoscopic television pictures, *Proceedings of the 1st International Festival of 3D Images* pp. 166–171.
- SULLIVAN, G. et BAKER, R. (1991), Rate-distortion optimized motion compensation for video compression, dans *IEEE Global Telecommunications Conference*, Vol. 1, pp. 85–90.
- THOMAS, L. et DERAÏ, F. (1993), Pruning of the transform space in block-based fractal image compression, dans *IEEE Proceedings of ICASSP: Image and Multi-Dimensional Signal Processing*, Vol. 5, pp. 341–344.
- THOMAS, L. et DERAÏ, F. (1995), Region-based fractal image compression using heuristic search, *IEEE Transactions on Image Processing* 4(6), pp. 832–838.
- TSELGOV, J., HENIK, A., DINSTEIN, I. et RABANY, J. (1990), Performance consequences of two types of stereo picture compression, *Human factors* **32**, pp. 173–182.
- VINES, G. et HAYES, M. (1993), Adaptive IFS image coding with proximity maps, dans *IEEE Proceedings of ICASSP: Image and Multi-Dimensional Signal Processing*, Vol. 5, pp. 349–352.
- VLEESCHAUWER, D. D. (1991), Coding of the disparity field and the correction signal in predictive 3DTV coding, *Proceedings of the 1st International Festival of 3D Images* pp. 213–221.

- WALL, L. et KINSNER, W. (1993). A fractal block coding technique employing frequency sensitive competitive learning, dans *Proceedings of IEEE Communications, Computers and Power*, pp. 320–329.
- WAN, S. J., WONG, S. K. M. et PRUSINKIEWICZ, P. (1988). An algorithm for multidimensional clustering, *ACM Transactions on Mathematical Software* 2(14), pp. 153–162.
- WANG, Y., HSIEH, X., HU, J. et LEE, O. (1995). Region segmentation based on active mesh representation of motion: Comparision of parallel and sequential approaches, dans *Proceedings of the IEEE International Conference on Image Processing*, Vol. 2, pp. 185–188.
- WANG, Y. et LEE, O. (1994). Active mesh – a feature seeking and tracking image sequence representation scheme, *IEEE Transactions on Image Processing* 3(5), pp. 610–624.
- WIEGAND, T., LIGHTSTONE, M., CAMPBELL, T. et MITRA, S. (1995). Efficient mode selection for block-based motion compensated video coding, dans *Proceedings of the IEEE International Conference on Image Processing*, Vol. 2, pp. 559–562.
- WILSON, D. L., NICHOLLS, J. A. et MONRO, D. M. (1994). Rate buffered fractal video, dans *Proceedings of the ICASSP*, Vol. V, pp. 505–508.

- WOHLBERG, B. E. et JAGER, G. D. (1995), Fast image domain fractal compression by DCT domain block matching, *Electronics Letters* **31**(11), pp. 869–870.
- WOLFE, J. M. (1986), Stereopsis and binocular rivalry, *Psychological Review* **93**(3), pp. 269–282.
- WONG, K.-H., HSU, C.-H. et KUO, C. C. J. (1993), Fractal-based image coding with polyphase decomposition, dans *SPIE Proceedings, Visual Communication and Image Processing*, Vol. 2094, pp. 1207–1218.
- YOUNG, R. et KINGSBURY, N. (1993), Video compression using lapped transforms for motion estimation/compensation and coding, *optical Engineering* **32**(7), pp. 1451–1463.
- ZHAO, Y. et YUAN, B. (1994), Image compression using fractals and discrete cosine transform, *Electronics Letters* **30**(6), pp. 474–475.
- ZHENG, H. et BLOSTEIN, S. (1995), Motion-based object segmentation and estimation using the MDL principle, *IEEE Transactions on Image Processing* **4**(9), pp. 1223–1235.
- ZHU, Y. M., GAO, Y. et GOUTTE, R. (1991), Stereo image compression by the combined use of the DCT and binocular fusion and rivalry, *Proceedings of the 1st International Festival of 3D Images* pp. 203–211.

ZIEGLER, M., TENGLER, W. et TABELING, P. (1991). Influence of camera calibration on the coding of stereo sequences. *Proceedings of the 1st International Festival of 3D Images* pp. 183–190.

Annexe A

Revue Bibliographique: Compléments

Cette annexe vient compléter le chapitre 1 en présentant le détail de certaines méthodes existant dans la littérature aussi bien en codage par compensation du mouvement qu'en codage fractal. Finalement, pour ce qui concerne la théorie de la neutralisation, cette annexe détaille également certaines méthodes de codage asymétrique ainsi que les résultats obtenus selon diverses techniques de codage.

A.1 Méthodes de codage par compensation du mouvement

Les méthodes de codage par compensation du mouvement qu'on trouve dans la littérature peuvent être classées de plusieurs manières. Ainsi, nous pouvons distinguer les différents systèmes selon le domaine sur lequel ils considèrent le mouvement. En effet, plusieurs systèmes évaluent le mouvement dans des régions fixes, comme des blocs par exemple. Par opposition, d'autres systèmes segmentent les images en objets de formes arbitraires. Enfin, le mouvement peut être considéré comme un champ bi-dimensionnel complexe (un vecteur par pixel) défini sur l'image au complet.

Toutes les méthodes ont un point commun; le mouvement est utilisé pour estimer les images au récepteur et la différence entre ces images et les vraies images est transmise. Un codage par transformation est souvent utilisé pour la compression de cette erreur.

A.1.1 Autres méthodes de compensation du mouvement par blocs

Codage adaptatif par classification des blocs Certains travaux ont raffiné la méthode de compensation du mouvement par bloc décrite dans la section 1.1.3 en effectuant une classification des blocs de l'image et en les codant de manière différente selon leur classe. La classification se fait en utilisant le contenu du bloc évalué par sa

variance (Huang, Chen, Wang, Hsieh et Hsieh 1994) ou en se servant d'un critère de minimisation du rapport taux/distorsion (Wiegand, Lightstone, Campbell et Mitra 1995).

Blocs de taille variable Un des problèmes des techniques de compensation du mouvement par blocs de taille fixe est que le mouvement est supposé uniforme dans chacun des blocs. En effet, attribuer un seul vecteur à un bloc revient à supposer que la portion de la scène projetée en ce bloc subit un mouvement rigide translationnel fronto-parallèle. De plus, le fait de supposer que chaque bloc provient d'un seul objet dans la scène produit des erreurs aux niveaux des contours des objets. Ceci nous oblige à utiliser des blocs de petite taille ce qui aura pour effet de diminuer le taux de compression et de rendre le calcul des vecteurs de déplacement plus sensible au bruit. Certains travaux ont résolu ce problème en utilisant des blocs de taille variable. Les algorithmes fonctionnent généralement dans le sens "grossier à fin" en commençant avec un bloc de large taille. La décision de diviser un bloc en plusieurs sous-blocs peut être basée sur l'erreur de reconstruction globale de ce bloc (évaluée par l'erreur *EQM*) trop importante (Hang, Schilling et Puri 1987), ou à un critère plus élaboré comme la minimisation de l'entropie globale dans l'image (Dufaux et Moscheni 1995). Dans la dernière méthode, un bloc est divisé en quatre sous-blocs si la diminution de l'entropie de l'erreur de reconstruction causée par cette meilleure modélisation du mouvement est supérieure à l'augmentation de l'entropie causée par l'ajout de vecteurs à transmettre. En outre, il existe des algorithmes, menés dans le sens "fin

à grossier". permettant de fusionner plusieurs blocs en un seul afin d'optimiser une allocation de bits (Sullivan et Baker 1991).

Hélas, la compensation du mouvement par blocs de taille variable est une méthode très coûteuse en temps de calcul. De plus, elle se limite à des blocs carrés et ne permet pas la fusion des blocs adjacents par opposition aux techniques par régions polygonales (voir section A.1.2).

Méthodes réduisant les effets de bloc Parfois, l'image reconstruite présente des démarcations visibles à la frontière des blocs appelée effet de bloc. Cet artefact provient de l'attribution pour chaque bloc d'un vecteur de déplacement évalué de manière indépendante des blocs voisins. Ceci cause des discontinuités dans le champ de déplacement aux frontières des blocs. Plus le taux de compression augmente, plus la démarcation entre les blocs va être visible. En outre, le choix arbitraire de blocs rectangulaires aggrave l'effet de bloc à cause de la sensibilité accrue du système visuel humain aux détails horizontaux et verticaux.

Plusieurs méthodes ont été proposées pour réduire la visibilité de l'effet de bloc. Le concept de blocs superposés est ainsi proposé (Young et Kingsbury 1993). Ces blocs sont pondérés par une fonction de fenêtrage qui donne plus d'importance au centre du bloc qu'à sa bordure. Un pixel est alors reconstruit comme la somme pondérée de tous les pixels de l'image précédente et qui sont déplacés à sa position.

Par ailleurs, plusieurs chercheurs proposent d'effectuer un post-traitement sur les images reconstruites au décodeur (Nakajima et al. 1994).

A.1.2 Compensation du mouvement par régions

Pour réduire l'effet de bloc, il est efficace de faire coïncider les bordures des objets en mouvement avec les bordures des blocs. Ainsi, la démarcation entre les différents mouvements estimés (un par région) sera moins visible car elle correspondra à des zones d'occlusion de la scène.

Régions polygonales Dans ces techniques, plusieurs blocs adjacents sont réunis pour former des régions dites polygonales. La décision d'unir ou de diviser des régions se fait pour optimiser un critère. Ainsi, la décision d'attribuer un bloc à une région se fait en se basant sur la différence entre le vecteur de déplacement du bloc et la moyenne des vecteurs de la région (Bartolini, Cappellini, Mecocci et Vagheggi 1994). Par ailleurs, on subdivise une région si l'erreur de reconstruction qui lui est associée est grande (Leonardi et Chen 1994). Les deux critères précédents effectuent la segmentation sans tenir compte de l'ajout d'information de segmentation à transmettre à chaque fois qu'on subdivise une région. Il peut être plus pertinent de se baser plutôt sur une fonction de coût qui tient compte de la qualité des images et de la compression obtenue (Lee 1995).

Ces approches sont des compromis sur les approches basées-bloc. La forme des régions est limitée à des unions de blocs et le mouvement de ces régions est représenté par un modèle translationnel. En fait, elles ne font qu'essayer de diminuer l'effet de bloc.

Compensation par régions à modèle de mouvement Les tendances actuelles optent pour segmenter la séquence d'images et représenter le mouvement de chaque région par un modèle plus élaboré que le modèle translationnel.

Les différentes méthodes se basant sur ce concept diffèrent par leur manière d'aborder le problème de la segmentation. En effet, il est possible de baser les modèles de mouvement sur des régions constantes (des blocs (Papadopoulos et Clarkson 1993) ou des triangles et des quadrilatères (Nakaya et Harashima 1994)) sauf que ceci cause les mêmes problèmes que les techniques par bloc. Les chercheurs optent plutôt pour une segmentation des images dans le domaine spatial¹ ou temporel² (Dufaux, Moscheni et Lippman 1995, Bonnaud et Labit 1994, Pardàs, Salembier et Gonzalez 1994, Salembier, Torres, Meyer et Gu 1995, Zheng et Blostein 1995). Le problème majeur de ces méthodes provient de la dualité existant entre la segmentation et l'estimation du mouvement. En effet, pour pouvoir calculer le mouvement, il faut savoir sur quelle région l'estimer et pour pouvoir déterminer la segmentation, il faut savoir quels pixels ont un mouvement homogène et donc, il faut connaître leur mouvement. D'autre part, l'information de segmentation est très coûteuse à transmettre (de l'ordre de 1bit point du contour de la région (Leonardi et Chen 1994)). Ce problème peut être résolu en estimant la segmentation au récepteur, comme à l'émetteur, une image en retard (Bonnaud et Labit 1994).

¹en se basant sur les intensités lumineuses des images

²en se basant sur les vecteurs de déplacement

Grilles adaptatives Au lieu de représenter l'image par des pixels localisés sur une grille uniforme sur l'image, les noeuds de la grille sont placés selon le contenu de l'images. Ainsi, ils ont plus de densité aux discontinuités spatiales.

Ces méthodes divisent l'image en taches de formes différentes qui exigent de modéliser le mouvement selon un modèle plus complexe que le modèle translationnel (Wang, Hsieh, Hu et Lee 1995).

A.1.3 Compensation dense

Ces techniques n'imposent aucune restriction sur le mouvement de la scène. Un champ dense ($\vec{v} = (v_1, v_2)$) composé d'un vecteur de mouvement par pixel doit être estimé et codé. Les tendances actuelles optent pour faire ceci de manière récursive pour minimiser l'erreur de reconstruction.

Une description détaillée des algorithmes de codage par compensation du mouvement par regions et dense peut être trouvée dans (Diab et Cohen 1996).

A.2 Codage fractal

A.2.1 Compression d'images fixes

A.2.1.1 Réduction de la complexité du codeur

Les méthodes de compression d'images fixes par fractales basées sur une recherche exhaustive du meilleur bloc D entraînent un temps de calcul prohibitif. Il existe de

nombreuses techniques de réduction de la complexité du codeur.

Certaines se basent sur une classification des blocs qui permet de réduire les candidats blocs D valides car la recherche de l'appariement n'est menée que pour des blocs D et I de même classe. La classification peut être effectuée par l'analyse du contenu du bloc (moyenne, variance) (Fisher et al. 1994b), par un détecteur d'arêtes (Jacquin 1992) ou à l'aide d'un réseau de neurones (Wall et Kinsner 1993).

Certains travaux réduisent la complexité du codeur en utilisant des caractéristiques invariantes pour faciliter la recherche dans un espace multi-dimensionnel formé des images et de leurs transformées affines. Cette représentation peut être construite dans le domaine spatial (Gotting, Ibenthal et Grigat 1995) ou dans le domaine fréquentiel (Wohlberg et de Jager 1995) (voir la section A.2.1.2). Par contre, cette représentation invariante peut introduire des pertes d'information rendant le résultat obtenu sous-optimal (du point de vue qualité visuelle) par rapport à la recherche exhaustive.

D'autres méthodes transforment le problème d'appariement en un problème de recherche du meilleur voisin dans un arbre (Bani-Eqbal 1995, Saupe 1994b, Kawamata, Nagahisa et Higuchi 1994).

Enfin, la mise en correspondance des blocs I et D peut être apprise et effectuée par un réseau de neurones de type Kohonen à apprentissage compétitif non supervisé (Bogdan et Meadows 1992, Hamzaoui 1995).

Pour diminuer le temps de calcul au codeur, il est également possible de réduire le

domaine de recherche des blocs D . Limiter l'espace de recherche peut consister à établir une *carte de proximité* avec une technique de recherche en spirale centrée sur la position du bloc I courant et s'en éloignant (Vines et Hayes 1993).

Cependant, il faut garder à l'esprit que l'étape déterminante d'un bon codage par fractales réside dans la précision de l'approximation d'un bloc I par un bloc D . Il ne faudrait donc pas que la réduction de l'espace de recherche provoque une baisse de la qualité de l'approximation et donc de l'image décodée.

Une manière de pallier à la diminution du nombre de blocs appariables peut consister à augmenter la complexité des transformations effectuées sur les blocs D . Dans cette optique, des méthodes hybrides (Monro 1993) utilisent la Transformation de Bath qui consiste à effectuer une transformation massique f_i non plus affine mais d'ordre supérieur dépendante de la position des pixels.

Toutes les méthodes décrites ci-dessus sont basées sur une mise en correspondance des blocs D avec les blocs I ce qui nécessite un espace de recherche. Or, il existe des systèmes de codage par fractales qui fonctionnent sans effectuer d'appariement (Monro et Woolley 1994, Dudbridge 1994).

Enfin, d'autres méthodes approximent un bloc I par une combinaison linéaire de blocs D qui constituent une base de blocs fixes et adaptatifs (Gharavi-Alkhansari et Huang 1994a, Gharavi-Alkhansari et Huang 1994b, Lepsoy, Øien et Ramstad 1993, Kim et Park 1994).

Pour obtenir une description détaillée des méthodes de réduction de complexité, il

suffit de se référer à l'article de Saupe et Hamzaoui (1994a).

A.2.1.2 Transformation de domaine

Toutes les méthodes décrites jusqu'à présent se basent sur un partitionnement de l'image en blocs I dans le domaine spatial original. Or, dans de nombreux systèmes de compression par blocs, il s'est souvent avéré pertinent d'effectuer une transformation de domaine sur ces blocs et de travailler dans un espace fréquentiel pour des raisons de compaction d'énergie et donc de compression. La recherche du codage par fractales s'est donc orientée naturellement vers une adaptation des techniques décrites précédemment à des blocs transformés. Les transformations effectuées sur les blocs D pour pouvoir approximer précisément un bloc I ont lieu sur les coefficients après application de la TCD . Les transformations affines correspondant à divers types d'isométries sont donc exprimées dans le domaine fréquentiel (Bracewell, Chang, Wang et Wang 1993).

Certaines méthodes codent ainsi les blocs transformés par TCD , soit par codage fractal et transmission de l'erreur de prédiction (Zhao et Yuan 1994), soit encore par un algorithme hybride codage fractal/codage par transformée (Barthel, Schuttemeyer, Voyé et Noll 1994). Cependant, ces deux méthodes requièrent le calcul de la transformée inverse TCD au décodeur contrairement à la méthode de Wohlberg et de Jager (1995) qui retranscrit les paramètres de codage dans le domaine spatial avant de les envoyer dans le canal. En outre, leur méthode présente un codeur accéléré par la con-

struction d'une représentation invariante à la plupart des transformations de blocs (transformation massique et certaines isométries).

Parmi les méthodes hybrides, certains travaux (Ratnakar et al. 1994) ont montré que l'utilisation conjointe du codage fractal et de la transformation *TCD* peut s'avérer inférieur aux méthodes de codage par transformation *TCD* pure.

A.2.1.3 Améliorations

Dans un système général de compression d'images, chaque bloc I est approximé par une transformation contractive du bloc D sous le critère de minimisation de l'erreur *EQM*. Cependant, il pourrait être intéressant de tenir compte du Système Visuel Humain (*SVH*) dans la mesure de distorsion utilisée ce qui permettrait d'améliorer sensiblement la qualité visuelle de l'image décodée (Beaumont 1991, Moon, Son, Kim et Kim 1995, Lewis et Knowles 1992).

Segmenter l'image en blocs de taille variable permet d'améliorer aussi la qualité de l'image reconstruite en diminuant les effets de blocs et de tenir compte du contenu variable des régions de l'image. La segmentation la plus fréquemment rencontrée est la segmentation en arbre quaternaire (Ratnakar et al. 1994, Bogdan 1994*a*, Fisher et al. 1994*b*). Celle-ci fonctionne de "grossier à fin" et part généralement de blocs I de taille maximale 32×32 et cesse la segmentation à des blocs de taille 4×4 (car le codage fractal de blocs 2×2 n'a aucun intérêt du point de vue compression). D'autres travaux segmentent l'image de manière adaptative selon les détails qu'elle contient

(Liu, Marlow et Murphy 1993).

La contrainte de contractivité spatiale consiste à considérer des blocs candidats D de taille supérieure à celle des blocs I (en général double). Cependant, cette contrainte n'est pas nécessaire à la convergence de l'image décodée vers le point fixe voulu. En effet, cette dernière est assurée par la contractivité de la transformation massique effectuée sur les niveaux de gris (paramètre a de module inférieur à 1). Bedford et al. (1994) ont ainsi considéré des facteurs de contractivité de 2 et de 1. Ceci présente en outre comme avantage de coder plus fidèlement les hautes fréquences de l'image. Les résultats montrent que la version combinée permet d'aboutir à un meilleur débit binaire pour une qualité d'image similaire que lorsque l'on ne considère que la contractivité spatiale 2. Par contre, il s'avère mauvais de ne considérer que des mises en correspondance avec une contractivité spatiale de 1 car des hautes fréquences indues pourraient apparaître dans certaines zones de l'image. Il faut cependant noter que cette relaxation peut supprimer la propriété d'indépendance en résolution de l'image décodée. Toutes les méthodes décrites jusqu'ici utilisent le même genre de partitionnement par blocs de différentes tailles non superposés de l'image. Elles engendrent donc toujours des artefacts de type effet de blocs (voir section A.1.1). Pour pallier à cet inconvénient, il existe le même type de techniques que pour les méthodes par compensation du mouvement. Ainsi, d'autres types de partitionnement de l'image sont envisagés: partitionnement avec superposition des blocs (Reusens 1994), basé sur des triangles et des quadrilatères (Davoine, Svensson et Chassery 1995) ou basé

sur une fusion des blocs en régions polygonales (Thomas et Deravi 1995). Un partitionnement basé sur une segmentation par région physique de la scène (correspondant à un objet) n'a jamais été envisagé car, outre la complexité introduite pour trouver la bordure des régions et la transmettre, le codage par fractales est alors très difficile à concevoir.

Enfin, un post-traitement par lissage de l'image reconstruite (Bedford et al. 1994) permet d'améliorer la qualité globale. Ce lissage pondère l'influence des pixels voisins du pixel lissé selon une distribution gaussienne et est effectué à chaque itération du module de décodage (et non seulement sur l'image-point fixe obtenue à la sortie du décodeur).

A.2.1.4 Accélération du décodeur

Une des qualités majeures du codage par fractales réside dans un décodage itératif mais rapide. Une manière d'accélérer encore le décodage est de construire le ou les blocs D approximant le bloc I de telle sorte que la reconstruction de l'image originale se fera en un nombre très faible d'itérations (typiquement 5) voire en une seule.

Le décodage rapide peut être assuré par un système d'orthogonalisation des blocs D (méthodes par combinaison de blocs de base) (Lepsoy et al. 1993, Kim, Kim et Lee 1995, Oien, Baharav, Lepsoy, Karnin et Malah 1994) ou par un espace de recherche causal des blocs D (Feig, Peterson et Ratnakar 1995) supprimant ainsi la nécessité d'itérer au décodeur.

A.2.2 Extension au codage de séquences vidéo

Les recherches actuelles du codage par fractales s'orientent vers le codage de séquences vidéo car le fort potentiel de compression laisse espérer que des séquences d'images puissent être codées avec une grande efficacité de codage en temps quasi-réel³. Les résultats les plus prometteurs dirigent naturellement les recherches vers les systèmes de codage à très bas débit et vers les applications de vidéo-conférence dans lesquelles les méthodes de codage par fractales surpassent les autres méthodes existantes.

Il existe globalement deux grands types de méthodes: celles qui fonctionnent avec des blocs bi-dimensionnels et celles qui étendent les résultats obtenus en codage d'images fixes en utilisant des blocs ou des régions tri-dimensionnelles.

A.2.2.1 Méthodes bi-dimensionnelles

Les méthodes bi-dimensionnelles consistent à coder une trame en utilisant soit le contenu de cette trame (codage intra-trame) soit celui de la trame précédente (codage inter-trame). Dans les deux cas, les blocs considérés sont bi-dimensionnels et la dimension temporelle n'est pas considérée dans la structure du bloc.

En 1992, Hurd, Gustavus et Barnsley (1992) ont présenté une nouvelle méthode de compression vidéo par fractales. La première trame de la séquence est codée par codage fractal. Puis, les trames suivantes sont codées à partir de l'image recons-

³pour l'instant, la méthode la plus rapide (Dudbridge 1994) obtient un résultat d'environ 18 trames/s

truite précédente ce qui limite temporellement la propagation des erreurs. Les trames décodées ont comme expression:

$$\hat{F}_1 = T^N(F_1) \simeq F_1$$

$$\hat{F}_n = T(\hat{F}_{n-1}) \simeq F_n \text{ pour } n \geq 2$$

Nous pouvons remarquer que le décodage (sauf pour la première trame) est non-itératif. Ce système possède les avantages de générer des rapports de compression élevés (de l'ordre de 80:1), d'avoir un décodage en temps réel et une bonne insensibilité aux erreurs.

Depuis cette méthode, de nombreux algorithmes ont été proposés. Ils regroupent généralement un ou plusieurs des éléments suivants:

Estimation des blocs actifs Chaque trame est partitionnée en blocs rectangulaires et chaque bloc est comparé au bloc situé à la même position dans la trame précédente reconstruite pour déterminer au moyen d'un différenciateur d'image les blocs qui ont été altérés (par mesure de distorsion). Ceci permet de réduire grandement le nombre de blocs I à considérer en ne transmettant pas d'information sur les zones de l'image inchangées entre deux trames consécutives (Bogdan 1994*b*, Fisher et al. 1994*b*).

Codage intra-trame versus codage inter-trame Le codage intra-trame consiste à coder chaque image de la séquence selon une des méthodes de compression des images fixes. Par opposition, le codage inter-trame utilise d'autres images de la séquence (généralement la précédente) pour coder l'image courante par un codage chaîné suivant l'équation A.1. Les méthodes bi-dimensionnelles combinent ou non de façon hybride ces deux types de codage.

Certains travaux codent la séquence de manière intra-trame (Monro et Nicholls 1994, Wilson, Nicholls et Monro 1994, Bogdan 1994*b*) et d'autres de manière inter-trame¹ (Fisher et al. 1994*b*). Les méthodes inter-frames présentent l'avantage d'être causales ce qui se traduit par un décodage non itératif et de ne pas avoir à respecter la contrainte de contractivité.

Évolution vers des méthodes hybrides Nous avons vu que les méthodes bi-dimensionnelles n'exploitent pas formellement la dimension temporelle des séquences vidéo. Or, la redondance temporelle est une propriété très importante qui peut être exploitée en combinant de manière hybride le codage fractal avec la compensation du mouvement. Certaines méthodes effectuent un codage par compensation du mouvement particulier dans lequel le modèle de mouvement choisi correspond à une modélisation fractale (Li et Forchheimer 1995, Fuh et Maragos 1991). Ainsi, elles généralisent les modèles de compensation du mouvement en considérant des transformations massiques et géométriques de type fractale qui ne représentent plus rien

¹la première image de la séquence est toujours codée de manière intra-trame

physiquement, mais permettent de raffiner la qualité obtenue. D'autres méthodes combinent véritablement la compensation du mouvement et le codage fractal.

Ainsi, certaines méthodes (Ali et al. 1992) effectuent un codage par bloc avec tentative de codage successivement par compensation du mouvement généralisé, codage fractal ou codage par transformée.

Enfin, d'autres travaux (Hürtgen et Büttgen 1993, Hürtgen et Stiller 1993, Andonova et Popovic 1994, de Faria et Ghanbari 1995) utilisent le codage fractal pour coder l'erreur de prédiction générée par un module de compensation du mouvement. Afin d'augmenter la qualité de reconstruction, la vitesse de codage et la compression, des éléments complémentaires, comme une recherche hiérarchique du livre de code et une segmentation multi-niveaux des blocs sont ajoutés Hürtgen et Stiller (1993).

Toutes ces méthodes donnent de bons résultats du point de vue qualité et compression, mais sont souvent coûteuses en temps de calcul.

A.2.2.2 Méthodes tri-dimensionnelles

Il existe des méthodes basées-blocs et des méthodes basées-régions à modèle de mouvement. Dans chaque cas, chaque image de la séquence est partitionnée de manière fixe ou adaptative en blocs ou régions tri-dimensionnels. Ces derniers sont également appelés blocs ou régions spatio-temporels. Ils comportent deux composantes spatiales en x et en y selon les deux dimensions de l'image et la troisième composante temporelle regroupant plusieurs trames successives de la séquence.

Les méthodes basées-blocs à partitionnement fixe (Barthel et Voyé 1995, Li, Novak et Forchheimer 1993) sont bien souvent des adaptations de méthodes de codage d'images fixes à des séquences vidéos utilisant des blocs tri-dimensionnels de taille fixe selon l'axe temporel (entre 2 et 5 images).

Certaines méthodes utilisent une analyse visuelle de la séquence pour choisir la méthode optimale de codage. Ainsi, pour les méthodes basées blocs tri-dimensionnels adaptatifs (Reusens 1993, Lazar et Bruton 1994), chaque bloc peut être codé soit en utilisant les similarités propres spatiales ou en exploitant les redondances temporelles. La partition d'un bloc est effectuée en prenant en compte le contenu de la séquence et la décomposition peut être menée selon l'axe temporel ou selon les axes spatiaux. Finalement, l'étape d'analyse peut être complètement séparée de l'étape de codage (Desknuydt, Desmet, Eycken et Oosterlinck 1994). L'analyse visuelle comporte une classification spatiale (selon la complexité du bloc) et temporelle (selon la dynamique du bloc) des blocs et permet de prévoir l'algorithme de codage local qui sera le mieux adapté à chaque bloc.

Dans les méthodes basées-régions, le partitionnement du volume d'images est effectué en utilisant l'estimation du mouvement. Ainsi, une région correspond à un objet déplacé entre chaque image. Le partitionnement va suivre le mouvement de cet objet au cours du temps (Baldine-Brunel et Hayes 1994).

Toutes ces méthodes donnent de très bons résultats du point de vue qualité et compression, particulièrement les méthodes basées-régions. Cependant, elles sont très

coûteuses en occupation mémoire et en temps de calcul.

Une description détaillée des différentes stratégies de codage d'images et de séquences d'images peut être trouvée dans (Frobert et Cohen 1996*a*).

A.3 Stéréoscopie et théorie de la neutralisation

A.3.1 Méthodes de codage utilisées

Outre la redondance intra-image, un système de codage de paires stéréoscopiques performant doit exploiter la redondance due à la géométrie stéréoscopique. Il existe deux grandes catégories de méthodes: symétriques et asymétriques.

Un survol des méthodes de codage symétriques peut être trouvé dans (Labonté et Laganière 1996). Ici, nous ne présentons que des méthodes de codage asymétrique qui exploitent la théorie de la neutralisation.

A.3.1.1 Codage de paires d'images stéréoscopiques

Il existe globalement deux types de méthodes de codage exploitant la théorie de la neutralisation. Certaines méthodes (Dinstein, Guy, Rabany, Tselgov et Henik 1989, Tselgov, Henik, Dinstein et Rabany 1990, Ziegler, Tengler et Tabeling 1991, Vleeschauwer 1991, Perkins 1992) se basent sur des appariements de blocs entre les deux images homologues et consistent donc à effectuer de la compensation de disparité tandis que les autres effectuent un sous-échantillonnage de l'image à faible résolution.

Compensation de disparité Une des deux images H est codée avec une méthode usuelle tandis que la compensation de disparité sert à coder l'image L à faible résolution. C'est l'analogie de la compensation du mouvement par blocs 1.1.3 adapté à la géométrie stéréoscopique.

D'autres méthodes combinent le codage prédictif et la compensation de disparité pour améliorer la qualité de reconstruction. Pour cela, en plus des vecteurs de disparité, ces méthodes codent et transmettent un signal de correction représentant l'erreur de prédiction entre l'image originale et l'image approximée. La détermination du signal de correction peut se faire dans le domaine image original (Ziegler et al. 1991, Vleeschauwer 1991) ou dans un domaine transformé (Perkins 1992).

Techniques par filtrage passe-bas/sous-échantillonnage Ces méthodes exploitent la théorie de la neutralisation en diminuant la résolution d'une des deux images à coder par un filtrage passe-bas et un sous-échantillonnage préalable au codage. Le module de filtrage/sous-échantillonnage peut être réalisé à l'aide d'une pyramide gaussienne (Dinstein et al. 1991, Dinstein, Kim, Tselgov et Henik 1989). Au récepteur, des techniques d'interpolation (Perkins 1992) sont effectuées pour reconstruire l'image à sa taille initiale.

Une pyramide gaussienne est une pyramide d'images dans laquelle chaque niveau de la pyramide g_k (sauf g_0 qui contient l'image originale) est une version filtrée passe-bas et sous-échantillonnée du niveau précédent.

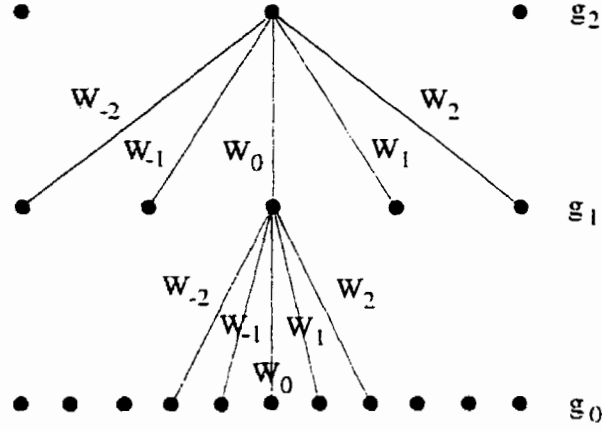


Figure A.1 : Pyramide gaussienne

Pour chaque pixel (i, j) de g_k , $0 < k \leq N - 1$,

$$g_k(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) g_{k-1}(2i + m, 2j + n) \quad (\text{A.1})$$

où N est le nombre de niveaux de la pyramide, $0 \leq i < C_k$, $0 \leq j < L_k$, où C_k et L_k sont le nombre de colonnes et de lignes du k -ième niveau.

$W(m, n)$ est un noyau générateur séparable défini par:

$$W(m, n) = W(m)W(n) \quad (\text{A.2})$$

où $W(0) = a$, $W(1) = W(-1) = \frac{1}{4}$, $W(2) = W(-2) = \frac{1}{4} - \frac{a}{2}$

Le schéma de la pyramide à 1 dimension est illustrée sur la figure A.1.

Au décodeur, l'image est reconstruite par interpolation de l'image g_k en utilisant la formule A.3. Soit $g_{k,q}$ le résultat de q interpolations de g_k et $g_{k,0} = g_k$.

$$g_{k,q}(i, j) = 4 \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) g_{k,q-1}\left(\frac{i-m}{2}, \frac{j-n}{2}\right) \quad (\text{A.3})$$

où $0 \leq k < N-1$, $0 \leq i < C_{k-q}-1$, $0 \leq j < L_{k-q}-1$, et seuls les termes pour lesquels $\frac{i-m}{2}$ et $\frac{j-n}{2}$ sont entiers sont inclus dans les sommations.

Il est à noter que $g_{k,k}$ a la même taille que g_0 .

A.3.1.2 Codage de séquences d'images stéréoscopiques

Pour ce qui est du codage asymétrique de séquences d'images stéréoscopiques, les méthodes existantes consistent à coder une séquence (celle à haute résolution) selon une méthode classique usuelle (par exemple de type MPEG 1.1.3) et à coder la séquence à faible résolution, filtrée passe-bas ou non par compensation du mouvement ou compensation de disparité (Schertz 1991).

A.3.2 Techniques d'expérimentations

Dispositifs d'affichage stéréoscopique Les stimulations visuelles (les deux images d'une paire stéréoscopique) auxquelles sont soumis les observateurs doivent être présentées en utilisant un dispositif spécial d'affichage stéréoscopique (Arditi 1986).

Les stéréogrammes sont réservés à la visualisation de paires d'images fixes et exigent une adaptabilité et une concentration du sujet parfois pénibles.

Un stéréoscope de type Wheatstone est un instrument qui, par un jeu de miroirs, permet de présenter devant chaque œil l'image qui lui correspond. Les deux directions

de fixation visuelles sont parallèles et la distance d'observation est d'environ 29cm. L'intervalle pupillaire est réglé entre 6 et 7cm.

Enfin, l'observateur peut porter des lunettes spéciales polarisantes et regarder un moniteur sur lequel s'affichent simultanément les deux images synchronisées d'une séquence stéréoscopique. Chaque verre des lunettes est polarisé de manière à ne laisser passer que l'image stéréoscopique correspondant au bon oeil.

Critères d'évaluation L'évaluation de la qualité de fusion peut être basée sur différents critères.

1. *Temps de réaction et Acuité visuelle*

Le temps de réaction est le temps mis pour choisir, entre deux objets, lequel est le plus proche de l'observateur. Plus ce temps est court, plus la perception de la profondeur est fine et donc meilleure est la qualité de la fusion. L'acuité visuelle est mesurée par la capacité d'identifier deux objets. Elle mesure donc la sensibilité aux détails de l'observateur.

Ces deux critères sont donc complémentaires pour évaluer la qualité d'un codeur stéréoscopique, tant du point de vue fusion que du point de vue résolution.

2. *Critères perceptuels*

Le critère perceptuel le plus simple et le plus couramment utilisé consiste en une échelle de classification de la qualité de la fusion et des dégradations en 5 points. Cependant, cette méthode présente le désavantage d'être difficilement

exploitable pour effectuer des comparaisons entre diverses séquences.

Aussi, il existe d'autres méthodes de type "choix forcé" où l'observateur se voit présenter deux séquences ou deux paires d'images stéréoscopiques et doit décider laquelle il préfère.

Enfin, il existe d'autres manières d'évaluer la qualité de fusion, soit par des questions formelles sur l'impression de profondeur et le contenu de la scène (Zhu et al. 1991), soit par un ajustement manuel du dispositif d'affichage. Dans ce dernier cas, les observateurs ajustent eux même la mise au point de l'affichage selon leur propre perception de la profondeur (Perkins 1992).

Une revue plus détaillée du concept de rivalité binoculaire ainsi que des différentes stratégies de codage asymétrique peut être trouvée dans (Frobert et Cohen 1996*b*).

Annexe B

Séquences d'images tests

Cette annexe contient la description (dans le tableau B.1) et la première image de chaque séquence test monoculaire ou stéréoscopique (à la figure B.1) utilisée pour effectuer l'étude de l'influence des paramètres et de la performance de notre système. En outre, elle contient la dernière image reconstruite pour chaque séquence test des expériences décrites dans le chapitre 4.

Les figures B.2 et B.3 représentent les images reconstruites de l'étude de comparaison de différents modes de codage: par compensation du mouvement seul, par codage fractal seul et par codage hybride présenté à la section 4.1.2.

Les figures B.4 et B.5 représentent les images reconstruites de l'étude de comparaison entre notre système de codage et la norme *MPEG* – 1 présentée à la section 4.1.3.

Enfin, la figure B.6, relative au codage de la séquence à basse résolution, contient les images reconstruites en utilisant diverses variantes de codage: avec filtrage de la

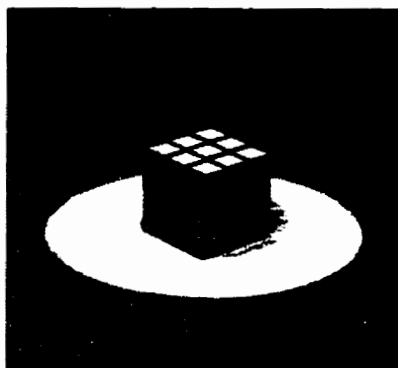
séquence et compensation du mouvement, avec filtrage de la séquence et compensation de disparité et sans filtrage de la séquence et avec compensation de disparité présentées à la section 4.2.2.

B.1 Description des séquences

Tableau B.1 : Description des séquences types sur lesquelles les tests ont été menés

Séquence	Type	Description du mouvement
"Rubic"	Mono	Rotation d'un cube et du socle qui le supporte
"Taxi"	Mono	Rotation d'un taxi situé au centre de l'image et translation de deux autres véhicules à gauche et à droite de l'image
"NASA"	Mono	Canette et son socle en mouvement globalement convergent vers la caméra
"Papillon"	Stéréo	Papillon ouvrant ses ailes dans un fond fortement texturé
"Tunnel"	Stéréo	Translation, rotation et divergence d'un train

B.2 Première image de chaque séquence originale



(a) Séquence "Rubic"



(b) Séquence "Taxi"



(c) Séquence "NASA"



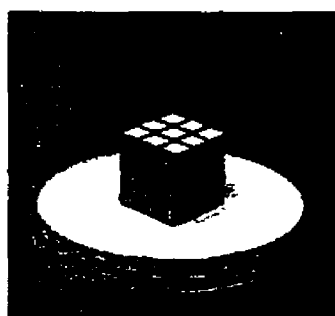
(d) Séquence "Papillon"



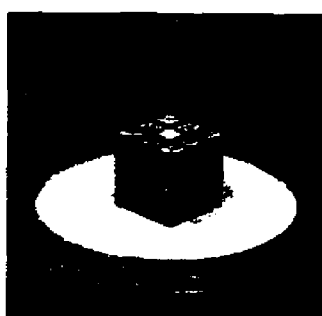
(e) Séquence "Tunnel"

Figure B.1 : Première image de chacune des séquences originales sur lesquelles les tests ont été menés

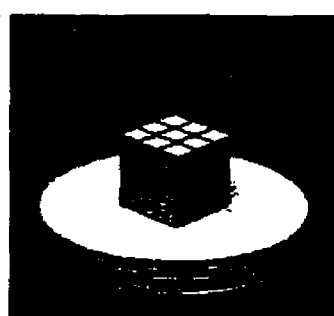
B.3 Dernière image reconstruite de chaque séquence



(a) "Rubik" avec codage fractal seul



(b) "Rubik" avec compensation du mouvement seul



(c) "Rubik" avec codage hybride



(d) "Taxi" avec codage fractal seul



(e) "Taxi" avec compensation du mouvement seul



(f) "Taxi" avec codage hybride



(g) "NASA" avec codage fractal seul



(h) "NASA" avec compensation du mouvement seul



(i) "NASA" avec codage hybride

Figure B.2 : Justification du mode hybride de codage

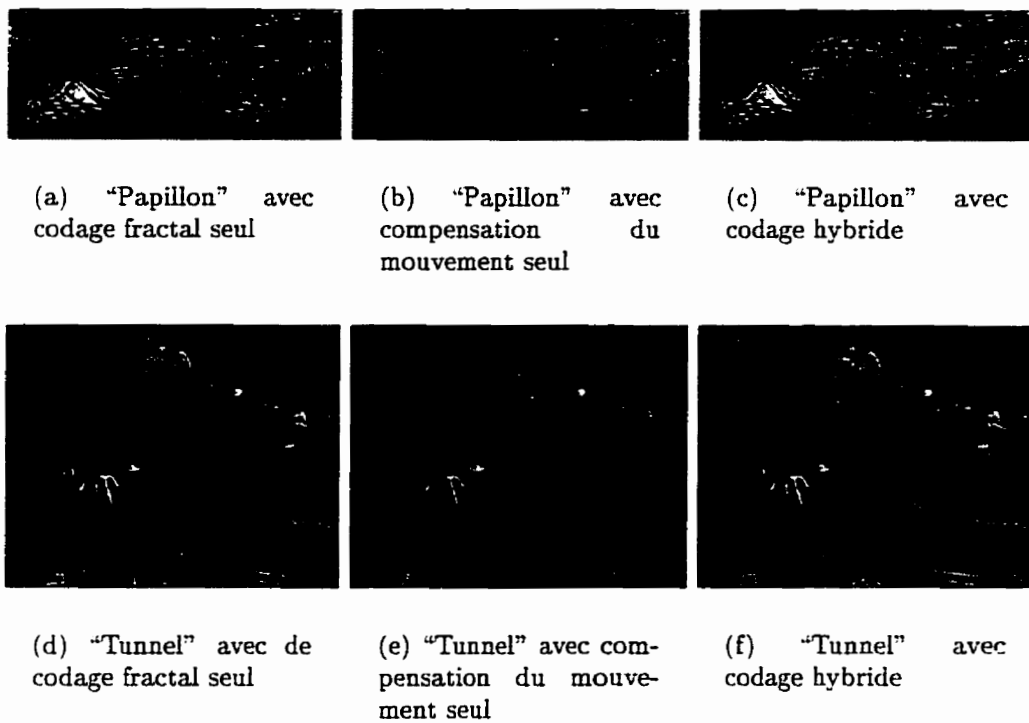


Figure B.3 : Justification du mode hybride de codage, suite

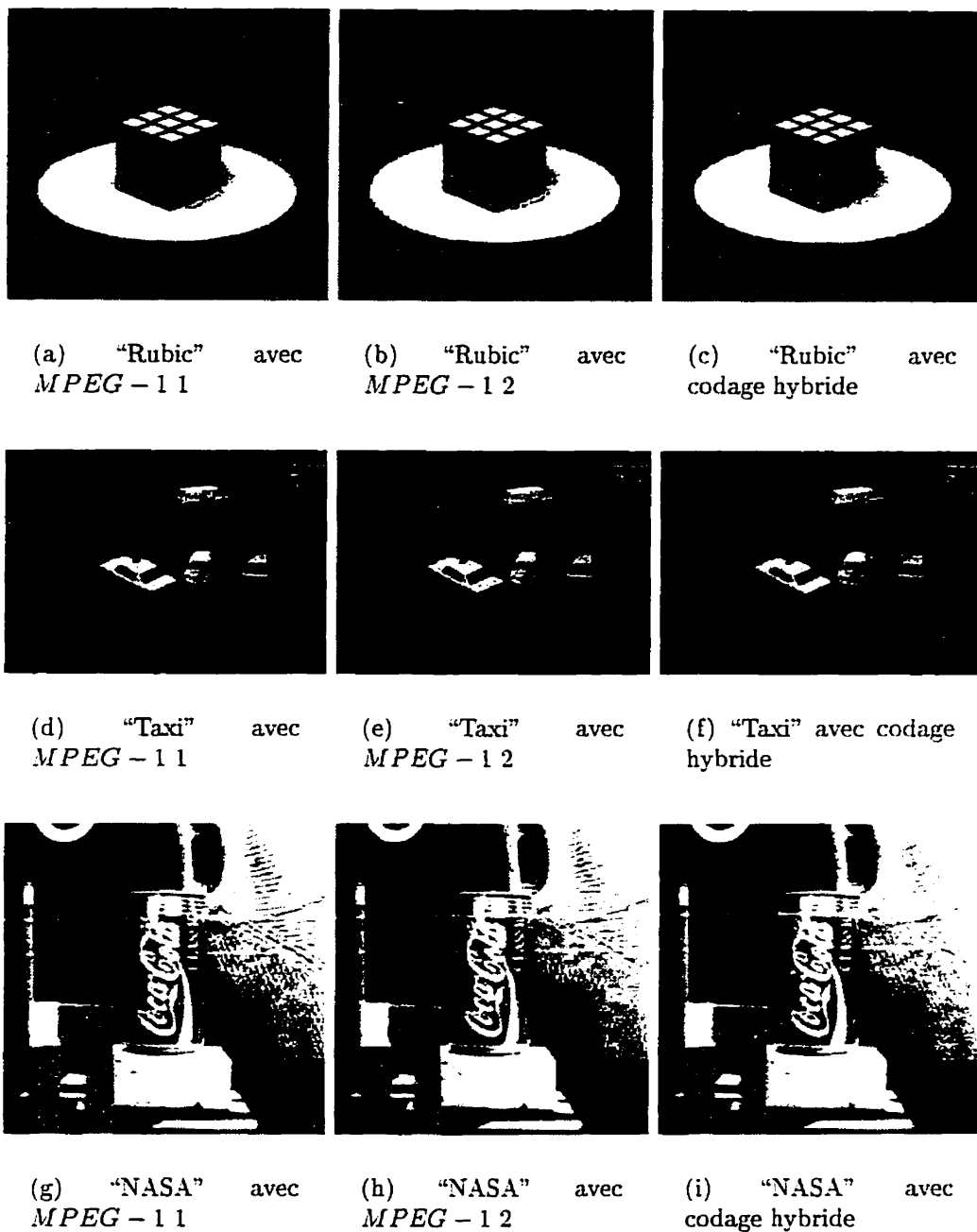


Figure B.4 : Comparaison du mode de codage hybride avec *MPEG - 1*

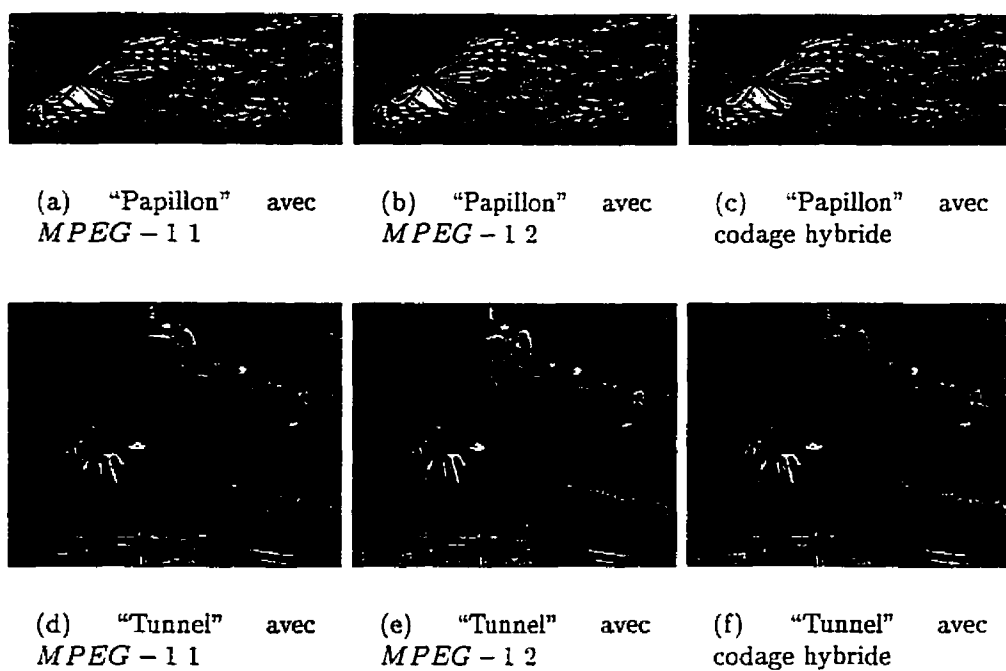


Figure B.5 : Comparaison du mode de codage hybride avec *MPEG - 1*. suite

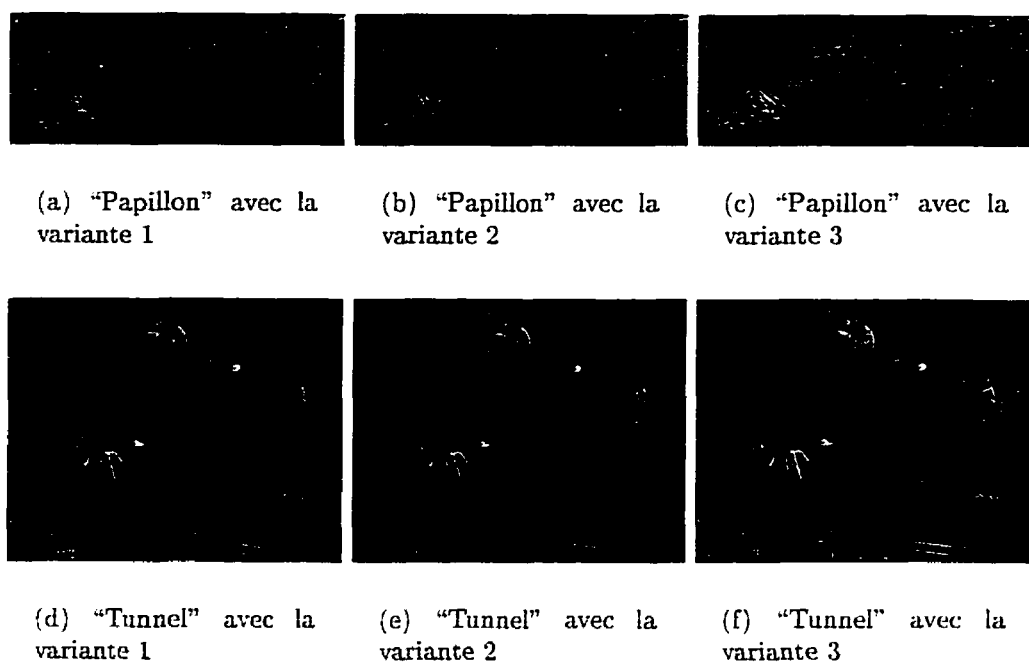


Figure B.6 : Comparaison entre les variantes de codage de la séquence à faible résolution

Annexe C

Construction d'une représentation invariante : Étude détaillée des compatibilités de signe

Cette annexe contient le détail des calculs nécessaires pour mener l'étude des impacts du masque présenté dans la section 3.1.3.5.

Le bloc à coder est noté \mathcal{I}^{trans} . Le bloc domaine candidat dont nous étudions la compatibilité avec \mathcal{I}^{trans} est noté \mathcal{D}^{trans} .

1. Calcul des termes prépondérants du masque :

Le but de cette étape est d'évaluer si les coefficients placés sur le masque du bloc \mathcal{I}^{can} comportent des termes prépondérants.

Soit Δ_{prep} avec $1 \leq \Delta_{prep} \leq \Delta_{prep}^{max}$, le nombre de coefficients placés sur le masque du bloc \mathcal{I}^{can} d'amplitude supérieure à Γ_{prep} .

Deux cas peuvent se présenter :

- $\Delta_{prep} = 0$ ou 1 , l'étude de compatibilité n'est pas nécessaire. Tous les blocs \mathcal{D}^{can} appartenant à λ_{frac} vont être candidats.
- $2 \leq \Delta_{prep} \leq \Delta_{prep}^{max}$, le choix du meilleur \mathcal{D}^{can} relatif au \mathcal{I}^{can} doit être conditionné à l'étude de compatibilité des signes de ces Δ_{prep} termes dans \mathcal{D}^{trans} et \mathcal{I}^{trans} .

2. Calculs préliminaires à l'étude :

Lorsque $\Delta_{prep} \geq 2$, il est nécessaire d'examiner les signes des Δ_{prep} termes prépondérants dans le bloc \mathcal{I}^{trans} correspondant et dans chaque bloc \mathcal{D}^{trans} candidat.

Les Δ_{prep} termes sont classés par ordre décroissant d'amplitude. Soit i avec $0 \leq i \leq \Delta_{prep}$, le i -ème terme prépondérant de \mathcal{I}^{trans} .

Les informations suivantes sont déterminées à partir de la donnée de la position des i termes prépondérants et de la valeur du drapeau P_I :

(u_i, v_i) = les indices u et v du terme i dans le bloc \mathcal{I}^{trans} .

$C(i)$ = la classe (1,2 ou 3) du terme i dans le bloc \mathcal{I}^{trans} .

$S_I(i)$ = le signe du terme i dans le bloc \mathcal{I}^{trans} .

Si $P_I = 1$, il faut permuter les rôles de u et v .

Conformément au tableau 3.1, nous avons :

$$C(i) = \begin{cases} 1 & \text{si } u_i \text{ impair et } v_i \text{ pair.} \\ 2 & \text{si } u_i \text{ pair et } v_i \text{ impair.} \\ 3 & \text{si } u_i \text{ et } v_i \text{ impairs.} \end{cases} \quad (C.1)$$

Nous avons:

$$S_I(i) = \begin{cases} 0 & \text{si } \mathcal{I}^{trans}(u_i, v_i) > 0. \\ 1 & \text{sinon.} \end{cases} \quad (C.2)$$

Soit \mathcal{D}^{trans} un bloc domaine candidat pour approximer \mathcal{I}^{trans} . Il est à noter que les positions des termes prépondérants de \mathcal{I}^{trans} dans le bloc \mathcal{D}^{trans} sont les mêmes à la transposition de u et v près. L'information P_D nous permet de calculer l'information suivante :

$$S_D(i) = \text{le signe du terme } i \text{ dans le bloc } \mathcal{D}^{trans}.$$

Si $P_D = 1$, il faut permuter les rôles de u et v .

3. Compatibilité des signes des blocs : *Définition* : Un bloc $B1$ présente Δ_s différences de signe inter-classe avec un autre bloc $B2$ si, pour Δ_s classes ($\Delta_s = 0, 1, 2, 3$), les signes attribués à chacune des trois classes diffèrent entre les deux

blocs.

Deux types de tests vont être utilisés :

(a) un test dit **INTRA-CLASSE** :

Il est mené de manière indépendante sur chacune des trois classes. Il consiste à étudier les compatibilités des signes de tous les termes prépondérants des blocs \mathcal{D}^{trans} et \mathcal{I}^{trans} appartenant à une même classe.

Soit Δ_{prep}^k le nombre de termes prépondérants de \mathcal{I}^{trans} appartenant à la même classe k .

Pour que le bloc \mathcal{D}^{trans} soit compatible avec le bloc \mathcal{I}^{trans} , il faut que la condition suivante soit vraie :

$$S_D(j) = \begin{cases} S_I(j) & \text{ou} \\ -S_I(j) & \text{pour } j = 1, \dots, \Delta_{prep}^k \text{ et } k \text{ non vide.} \end{cases} \quad (C.3)$$

En cas de réussite de ce test et si les termes prépondérants se répartissent dans les trois classes, chaque classe des deux blocs se voient attribuer un signe.

Chaque classe k avec $k = \{1, 2, 3\}$ d'un bloc \mathcal{D}^{trans} est considérée dans son ensemble. Nous lui associons le signe "-" s'il a fallu modifier les signes de tous les termes contenus dans cette classe—lors du test intra-classe—et le signe "+" si aucun terme ne devait être modifié. Ces deux cas sont les

Bloc I^{trans}	Bloc D^{trans}	Δ_s : nombre de différences de signes	Compatibilité						
<table border="1"><tr><td>+</td><td>+</td><td>+</td></tr></table>	+	+	+	<table border="1"><tr><td>+</td><td>+</td><td>+</td></tr></table>	+	+	+	0	oui
+	+	+							
+	+	+							
<table border="1"><tr><td>-</td><td>+</td><td>-</td></tr></table>	-	+	-	<table border="1"><tr><td>+</td><td>-</td><td>-</td></tr></table>	+	-	-	1	non
-	+	-							
+	-	-							
<table border="1"><tr><td>+</td><td>+</td><td>-</td></tr></table>	+	+	-	<table border="1"><tr><td>+</td><td>-</td><td>-</td></tr></table>	+	-	-	2	oui
+	+	-							
+	-	-							
<table border="1"><tr><td>-</td><td>+</td><td>-</td></tr></table>	-	+	-	<table border="1"><tr><td>-</td><td>-</td><td>-</td></tr></table>	-	-	-	3	non
-	+	-							
-	-	-							

Signes <i>intra-classe</i> des 3 classes du bloc:	1	2	3
	X	X	X

Figure C.1 : Compatibilité des signes pour 3 classes différentes

seuls à considérer car le bloc D^{trans} a réussi le premier test intra-classe.

Les 3 classes du bloc I^{trans} reçoivent le signe "-".

La figure C.1 illustre les différents cas possibles.

Il est à noter que les trois classes jouent un rôle parfaitement symétrique.

Donc, tous les cas possibles sont considérés en permutant les numérotations des classes.

(b) un test dit **INTER-CLASSES** :

Il consiste à comparer les compatibilités de signes des classes des deux blocs entre elles. En effet, lorsque les termes prépondérants se répartissent dans les trois classes possibles, des problèmes peuvent se présenter à cause de l'inter-corrélation entre ces trois classes.

4 cas peuvent se présenter selon la valeur de Δ_s :

i. $\Delta_s = 0$:

Dans ce cas, aucune classe de \mathcal{D}^{trans} ne doit voir son signe modifié car les signes de chaque classe des blocs \mathcal{I}^{trans} et \mathcal{D}^{trans} se superposent déjà parfaitement. Les deux blocs sont donc compatibles.

ii. $\Delta_s = 1$:

Pour une parfaite superposition des signes des 3 classes des deux blocs, il faudrait donc modifier le signe d'une classe sans affecter les deux autres.

Par exemple, d'après la figure C.1, il faut appliquer sur la classe 3 une isométrie contenant le terme $(-1)^u$ ou $(-1)^v$ ce qui affectera automatiquement la classe 1 ou 2.

La modification du signe d'une seule classe est donc impossible. Les deux blocs sont incompatibles.

iii. $\Delta_s = 2$:

Pour une parfaite superposition des signes des 3 classes des deux blocs, il faudrait donc modifier le signe de deux classes sans affecter la troisième.

Par exemple, d'après la figure C.1, il faut appliquer sur la classe 2 une isométrie contenant le terme $(-1)^v$. Cet opérateur affecte automatiquement la classe 3 sans toucher à la classe 1.

La modification est donc possible et, par symétrie, pour toutes les

combinaisons de classes. Les deux blocs sont donc compatibles.

iv. $\Delta_s = 3$:

Pour une parfaite superposition des signes des 3 classes des deux blocs, il faudrait donc appliquer la même isométrie pour les trois classes ce qui est impossible d'après le tableau 3.1. Les deux blocs sont donc incompatibles.

En résumé, pour vérifier la compatibilité, il suffit de compter le nombre de différences de signes Δ_s entre les deux blocs:

$$\mathcal{D}^{trans} = \begin{cases} \text{accepté} & \text{si } \Delta_s = \text{pair.} \\ \text{rejeté} & \text{sinon.} \end{cases} \quad (\text{C.4})$$

4. Exemple :

La figure C.2 montre quantitativement comment l'étude de compatibilité de signes permet de rejeter le bloc menant à un mauvais résultat dans l'exemple présenté sur la figure 3.8.

Dans cet exemple, nous avons fixé $P_I = P_{D_1} = P_{D_2} = 0$.

Le bloc \mathcal{I}^{trans} comporte 4 termes prépondérants. Il est donc nécessaire d'effectuer l'étude de la compatibilité des signes. En utilisant les expressions C.2 et C.1, nous obtenons les résultats suivants :

$$(u_1, v_1) = (1, 0) \quad S_I(1) = 0 \quad C_I(1) = 1.$$

$$(u_2, v_2) = (1, 1) \quad S_I(2) = 1 \quad C_I(2) = 3.$$

$$(u_3, v_3) = (0, 1) \quad S_I(3) = 1 \quad C_I(3) = 2.$$

$$(u_4, v_4) = (2, 1) \quad S_I(4) = 1 \quad C_I(4) = 2.$$

Pour les deux blocs \mathcal{D}_1^{trans} et \mathcal{D}_2^{trans} candidats, nous calculons selon C.2 :

$$S_{D1}(1) = 1 \quad S_{D2}(1) = 1.$$

$$S_{D1}(2) = 0 \quad S_{D2}(2) = 1,$$

$$S_{D1}(3) = 1 \quad S_{D2}(3) = 1.$$

$$S_{D1}(4) = 0 \quad S_{D2}(4) = 0.$$

Les quatre termes prépondérants sont de trois classes différentes. Il faut donc mener en premier lieu l'étude de compatibilité de signes intra-classe. Celle-ci réussit pour les deux blocs. En effet, seule la classe 2 comporte 2 termes dont les signes sont les mêmes pour \mathcal{D}_1^{trans} ou \mathcal{D}_2^{trans} que pour \mathcal{I}^{trans} .

Ensuite, pour l'étude de la compatibilité de signes inter-classes, nous devons calculer le nombre de différences de signes entre chaque classe de chaque bloc candidat \mathcal{D}_1^{trans} ou \mathcal{D}_2^{trans} et chaque classe du bloc \mathcal{I}^{trans} .

Nous obtenons :

$$\Delta_s(D1) = \text{pair},$$

$$\Delta_s(D2) = \text{impair}.$$

Par conséquent, d'après C.4, \mathcal{D}_1^{trans} est compatible avec \mathcal{I}^{trans} alors que \mathcal{D}_2^{trans} est rejeté.

I^{trans}	D_1^{trans}	D_2^{trans}																																				
<table border="1"> <tr><td>0</td><td>0.3</td><td></td></tr> <tr><td>-0.2</td><td>-0.3</td><td>0.1</td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table>	0	0.3		-0.2	-0.3	0.1							<table border="1"> <tr><td>0</td><td>-0.4</td><td></td></tr> <tr><td>-0.1</td><td>0.3</td><td>0.15</td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table>	0	-0.4		-0.1	0.3	0.15							<table border="1"> <tr><td>0</td><td>-0.3</td><td></td></tr> <tr><td>-0.2</td><td>-0.3</td><td>0.1</td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table>	0	-0.3		-0.2	-0.3	0.1						
0	0.3																																					
-0.2	-0.3	0.1																																				
0	-0.4																																					
-0.1	0.3	0.15																																				
0	-0.3																																					
-0.2	-0.3	0.1																																				

a) 4 termes prépondérants de classe:

	1	
2	3	2

b) Etude des signes des termes prépondérants

	+	
-	-	+

	-	
-	+	+

	-	
-	-	+

c) Compatibilité des signes intra-classe:

1	2	3
+	+	+

-	+	-
---	---	---

-	+	+
---	---	---

c) Nombre de différences de signes

2 -> D_1 compatible avec I 1 -> D_2 incompatible avec I

ACCEPTÉ

REJETÉ

Figure C.2 : Exemple d'étude de compatibilité de signes

Annexe D

Influence des différents paramètres: Courbes complémentaires

Cette annexe contient un ensemble de figures venant compléter celles figurant dans le chapitre 3. Elles représentent, pour les 5 séquences tests, l'effet de tous les paramètres utilisés dans le codeur sur le taux de compression et l'erreur quadratique moyenne considérés séparément et obtenus à chacune des 5 itérations.

D.1 Module de compensation du mouvement

Les figures D.1 et D.2 illustrent l'influence du seuil de succès du module Γ_{mvt} sur le taux de compression et l'erreur quadratique moyenne.

Les figures D.3 et D.4 illustrent l'influence de la longueur de la spirale λ_{mvt} sur le taux de compression et l'erreur quadratique moyenne.

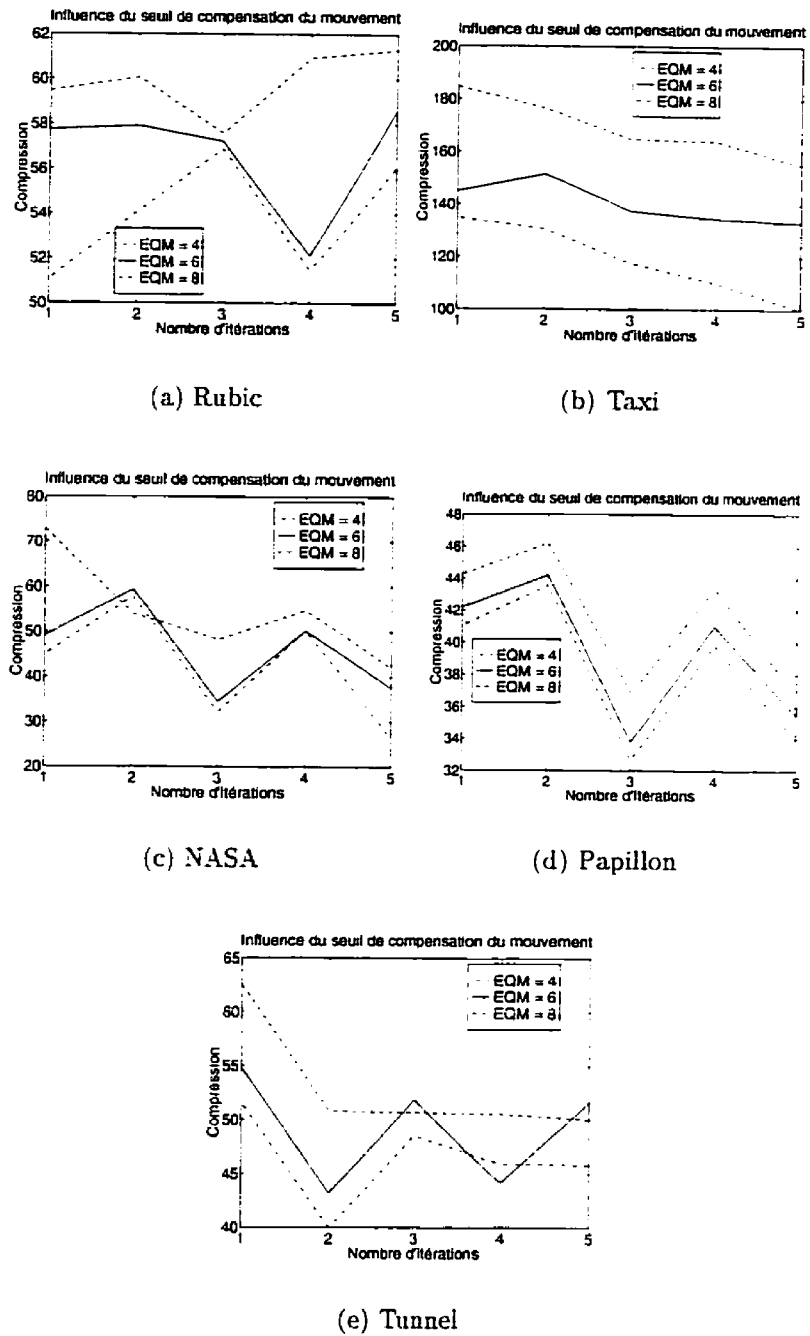
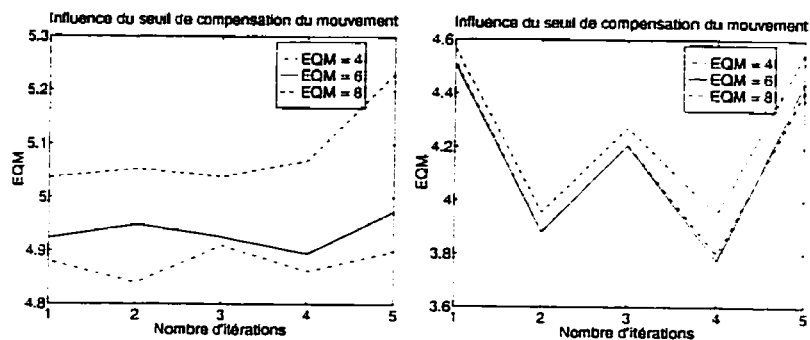
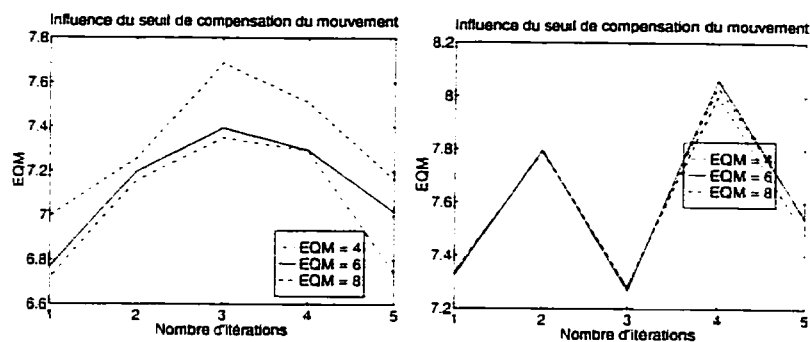


Figure D.1 : Influence du seuil de succès du module de compensation du mouvement Γ_{mut} sur le taux de compression



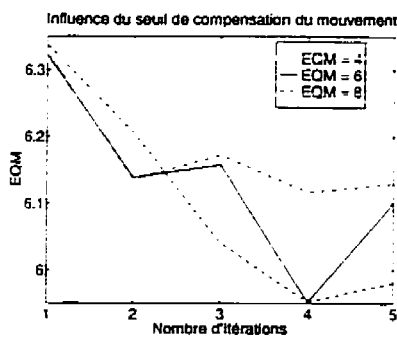
(a) Rubic

(b) Taxi



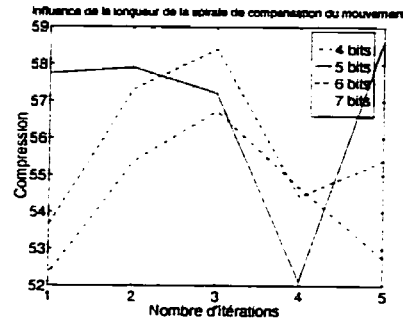
(c) NASA

(d) Papillon

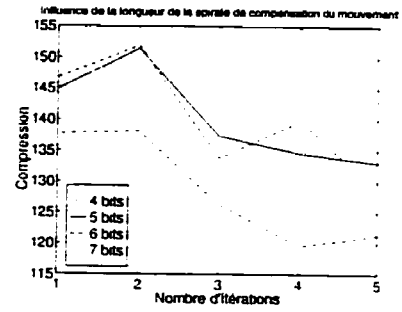


(e) Tunnel

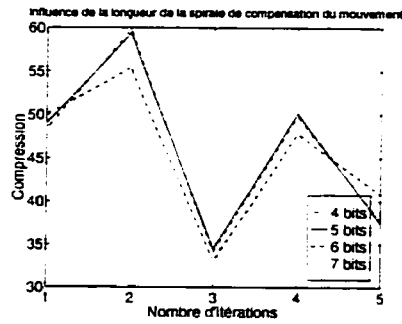
Figure D.2 : Influence du seuil de succès du module de compensation du mouvement Γ_{mut} sur l'erreur quadratique moyenne



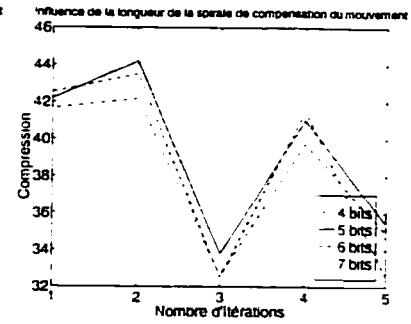
(a) Rubic



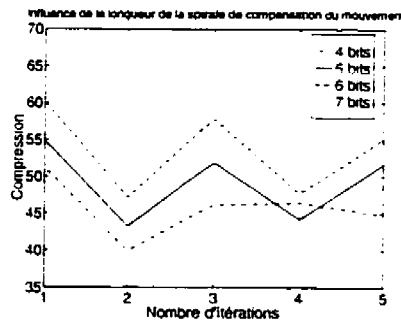
(b) Taxi



(c) NASA



(d) Papillon



(e) Tunnel

Figure D.3 : Influence de la longueur de la spirale de recherche λ_{mvt} sur le taux de compression

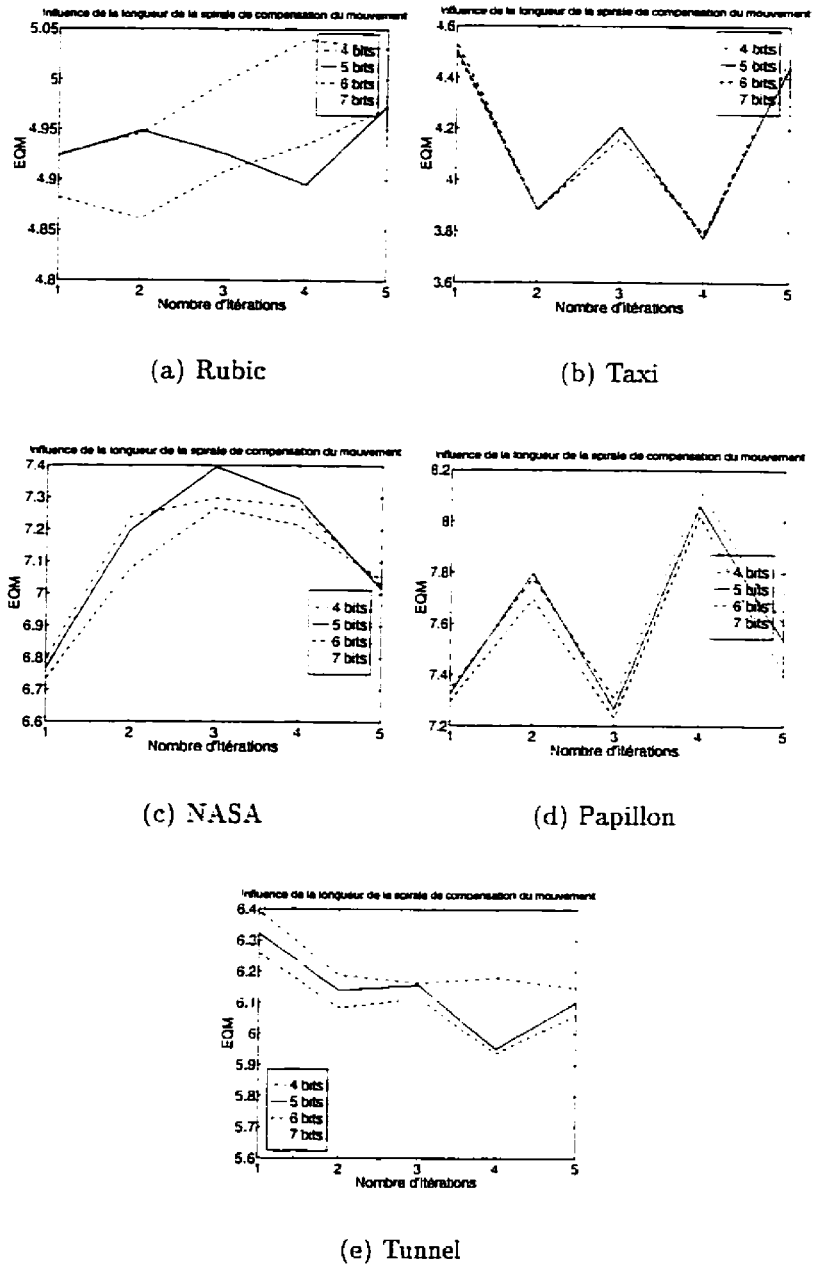


Figure D.4 : Influence de la longueur de la spirale de recherche λ_{mut} sur l'erreur quadratique moyenne

D.2 Module de codage fractal

Les figures D.5 et D.6 illustrent l'influence de la longueur de la spirale λ_{frac} sur le taux de compression et l'erreur quadratique moyenne.

Les figures D.7 et D.8 illustrent l'influence de la quantification du facteur d'échelle a sur le taux de compression et l'erreur quadratique moyenne.

Les figures D.9 et D.10 illustrent l'influence de la quantification du facteur de décalage b sur le taux de compression et l'erreur quadratique moyenne.

Les figures D.11 et D.12 illustrent l'influence combinée du nombre maximum de termes prépondérants Δ_{prep}^{max} et du seuil de prépondérance Γ_{prep} sur le taux de compression et l'erreur quadratique moyenne.

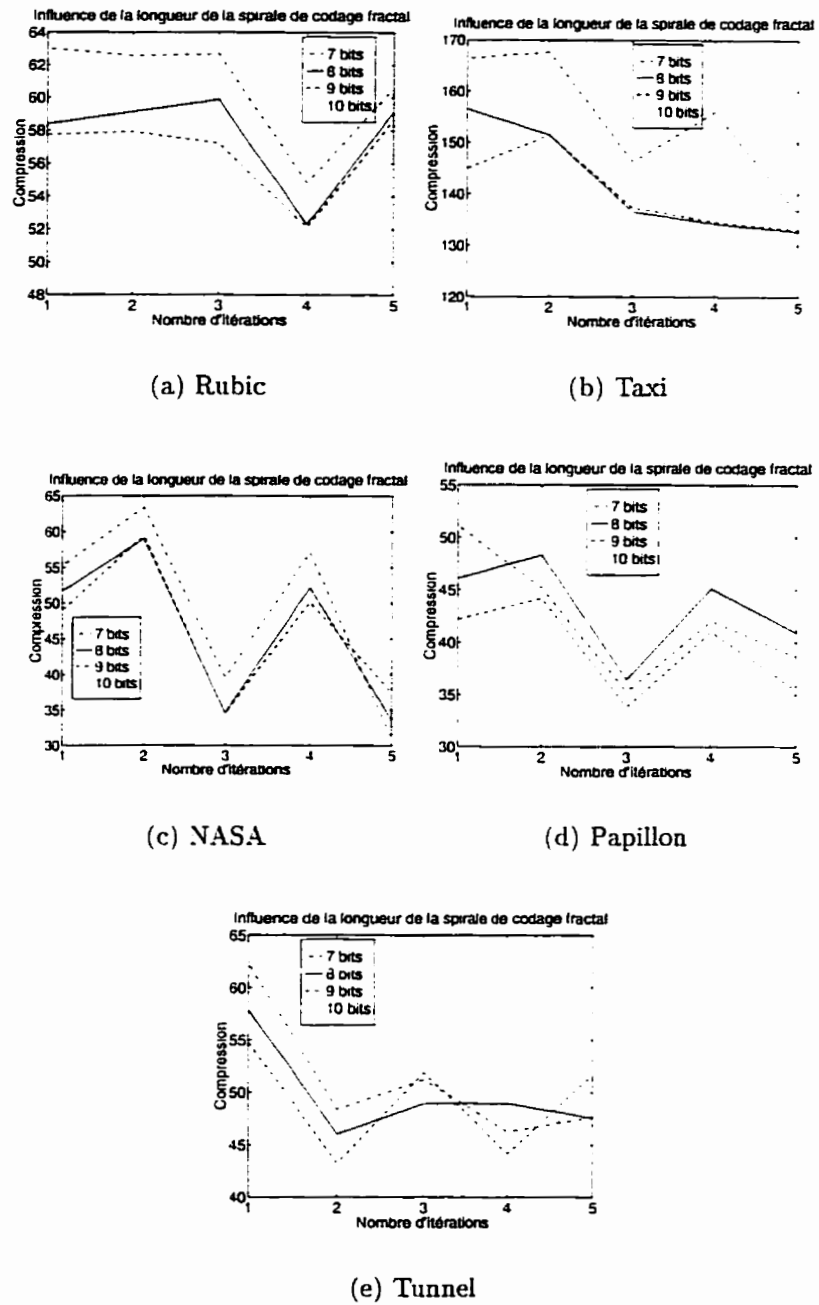


Figure D.5 : Influence de la longueur de la spirale de recherche λ_{frac} sur le taux de compression

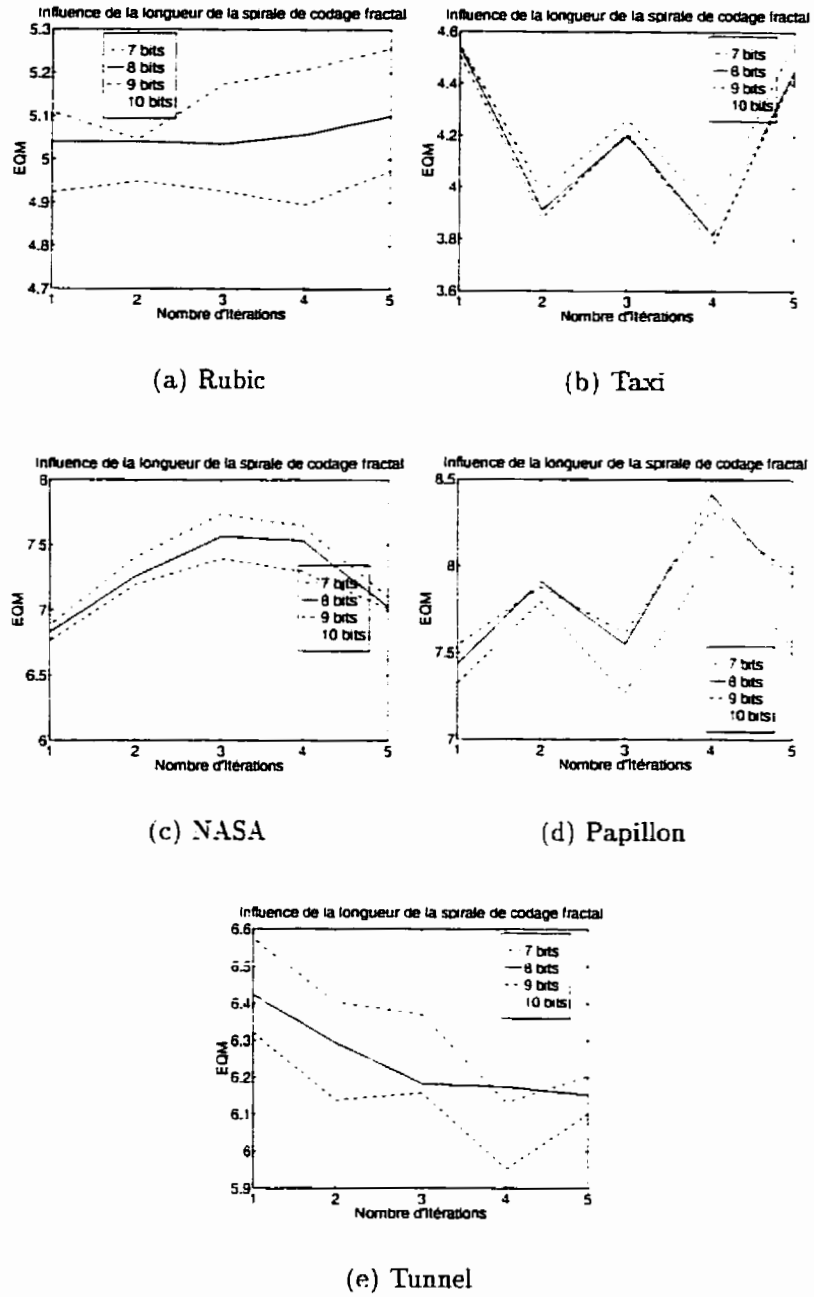


Figure D.6 : Influence de la longueur de la spirale de recherche λ_{frac} sur l'erreur quadratique moyenne

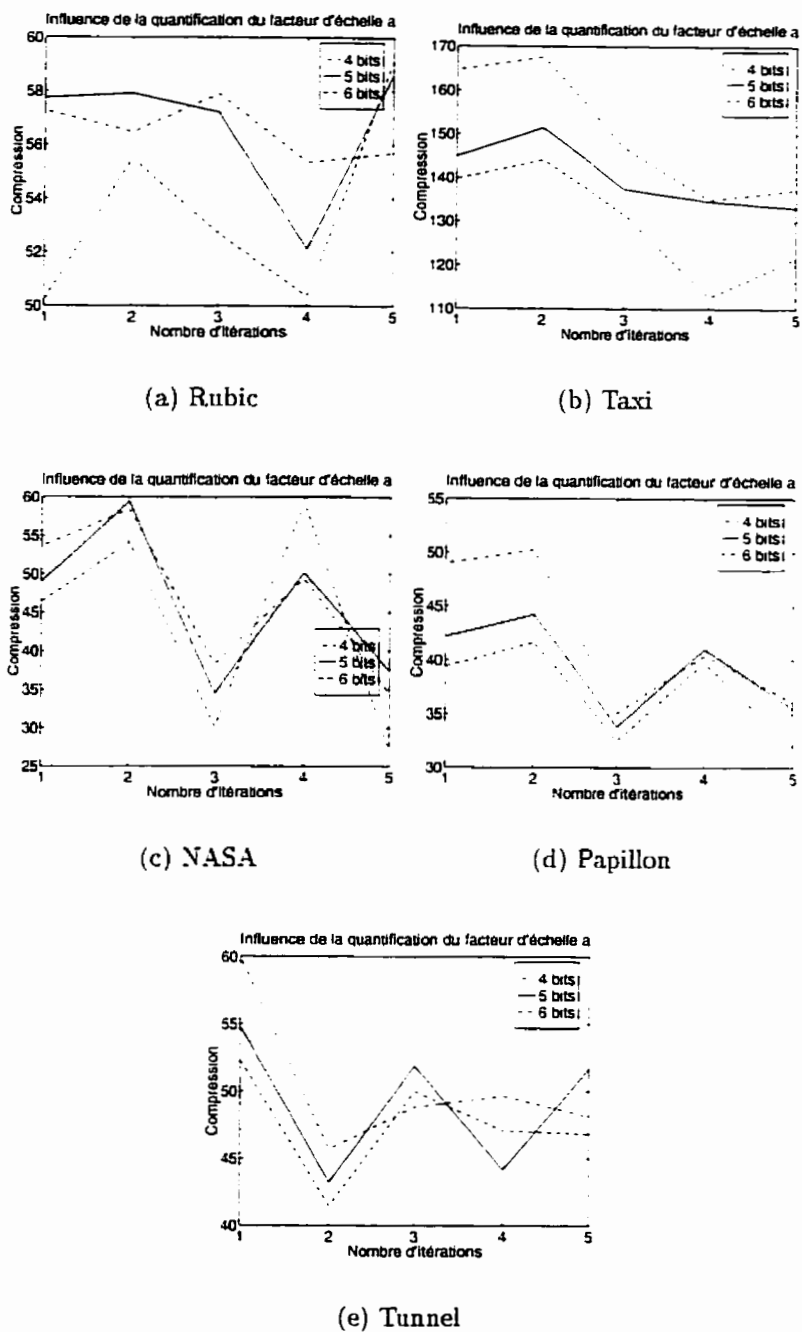
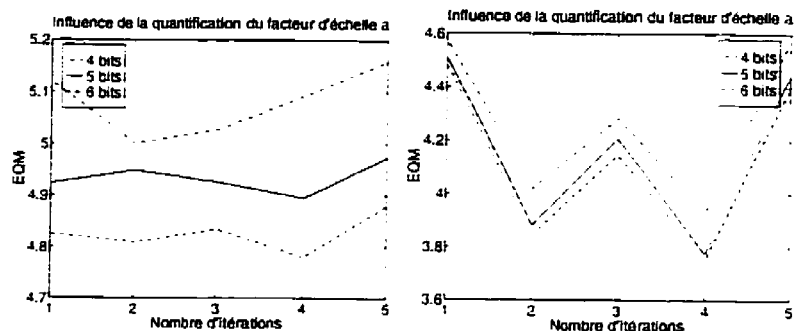
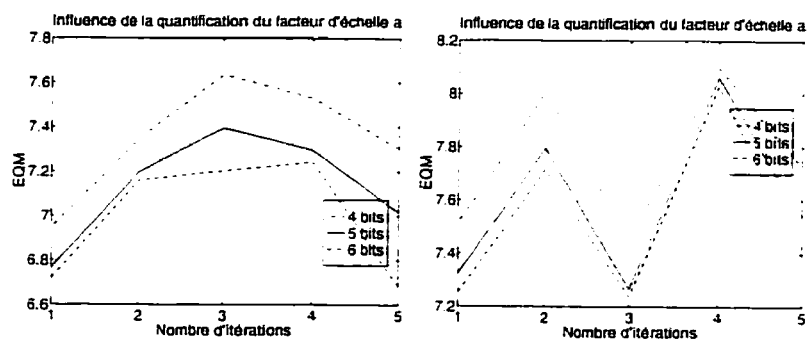


Figure D.7 : Influence de la quantification du facteur d'échelle α sur le taux de compression



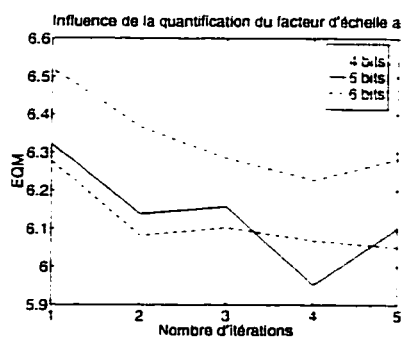
(a) Rubic

(b) Taxi



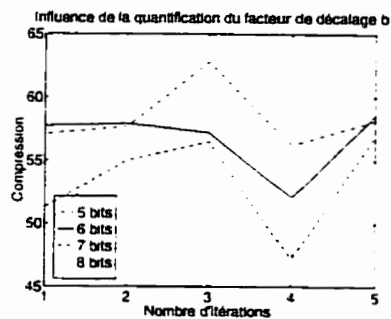
(c) NASA

(d) Papillon

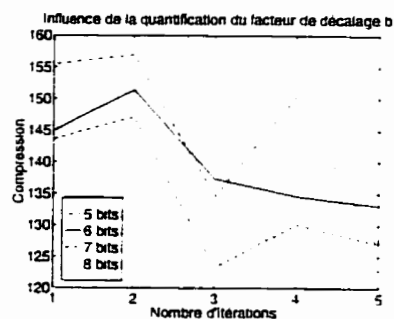


(e) Tunnel

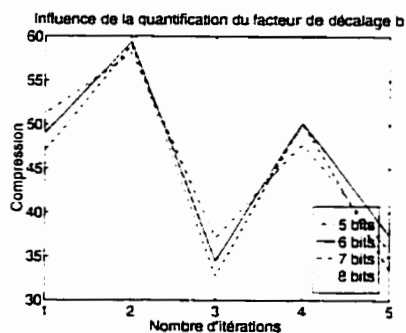
Figure D.8 : Influence de la quantification du facteur d'échelle α sur l'erreur quadratique moyenne



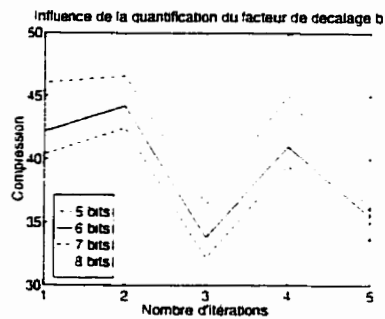
(a) Rubic



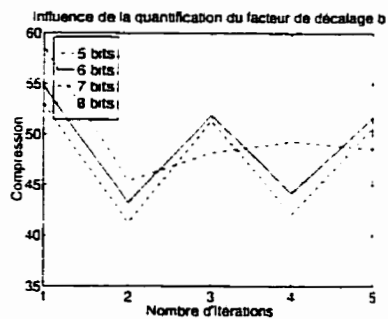
(b) Taxi



(c) NASA



(d) Papillon



(e) Tunnel

Figure D.9 : Influence de la quantification du facteur de décalage b sur le taux de compression

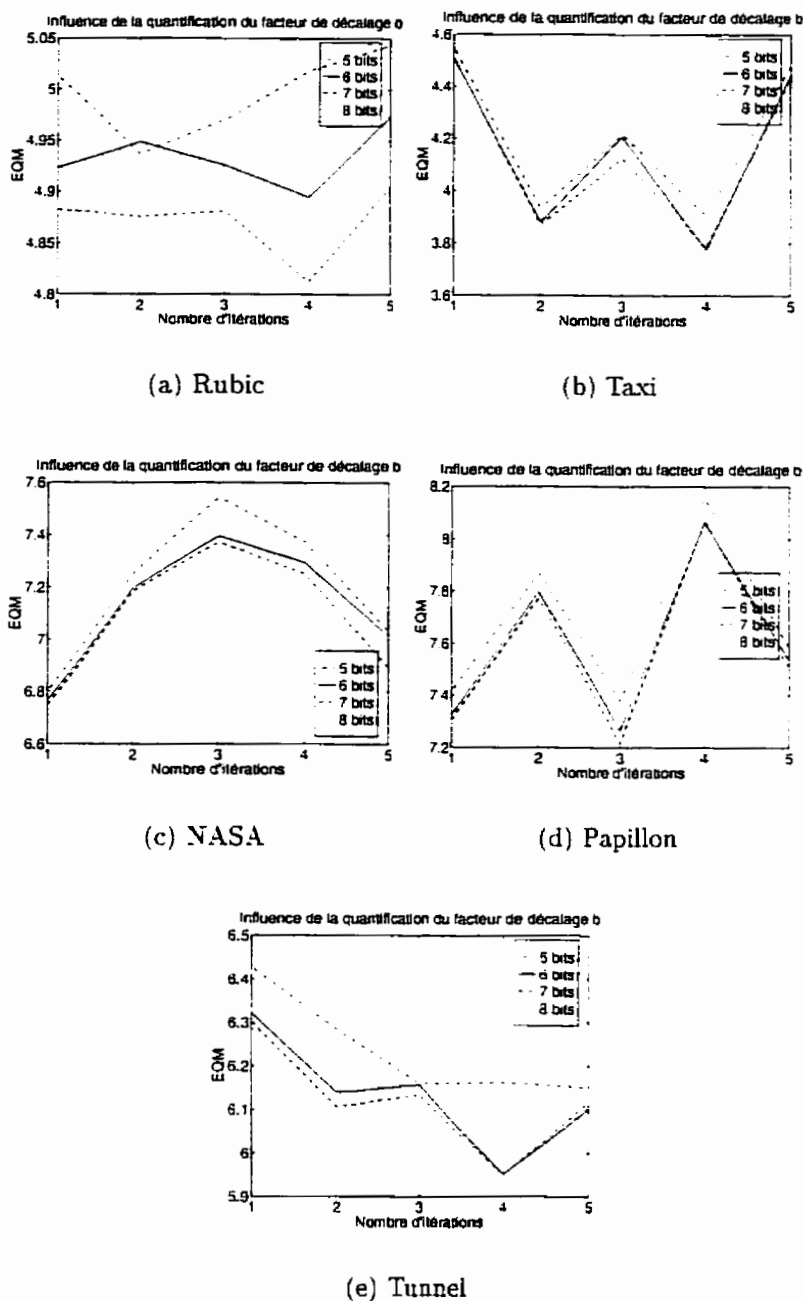


Figure D.10 : Influence de la quantification du facteur de décalage b sur l'erreur quadratique moyenne

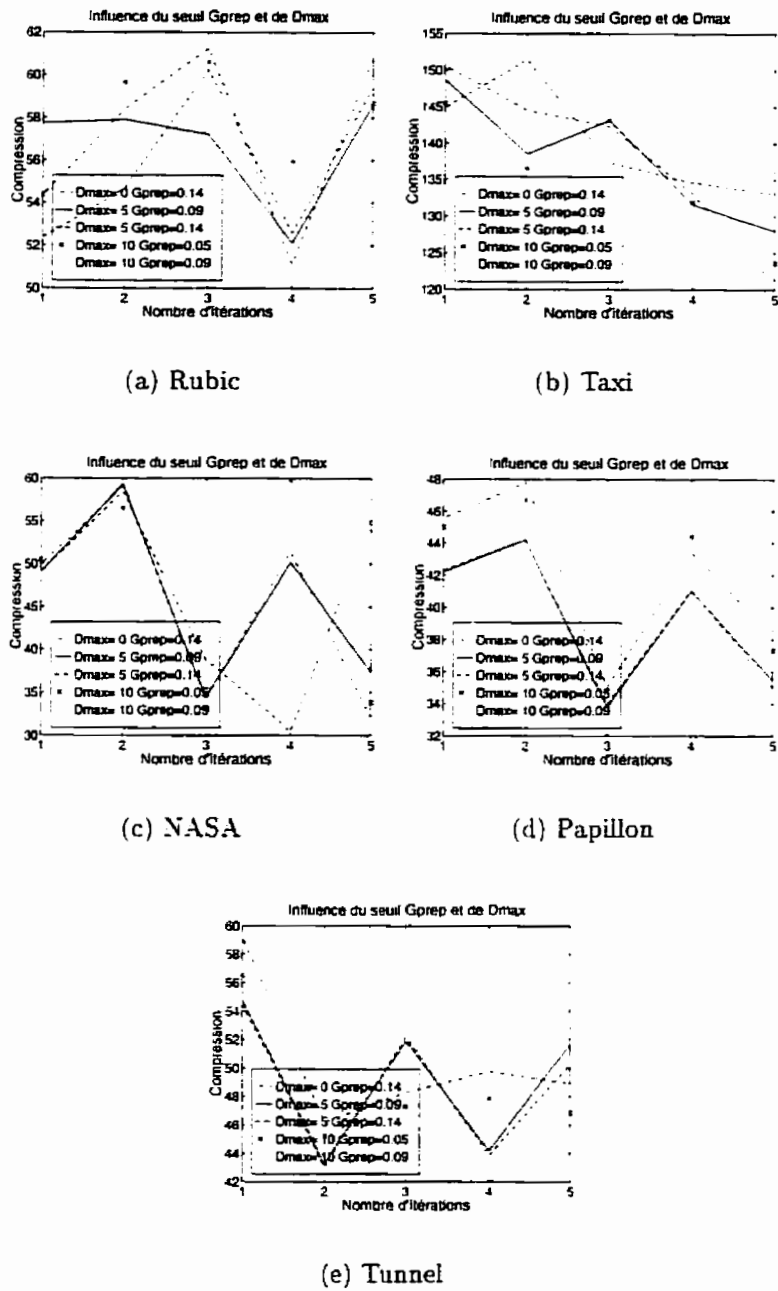


Figure D.11 : Influence du nombre maximum de termes prépondérants Δ_{prep}^{max} et du seuil de prépondérance Γ_{prep} sur le taux de compression

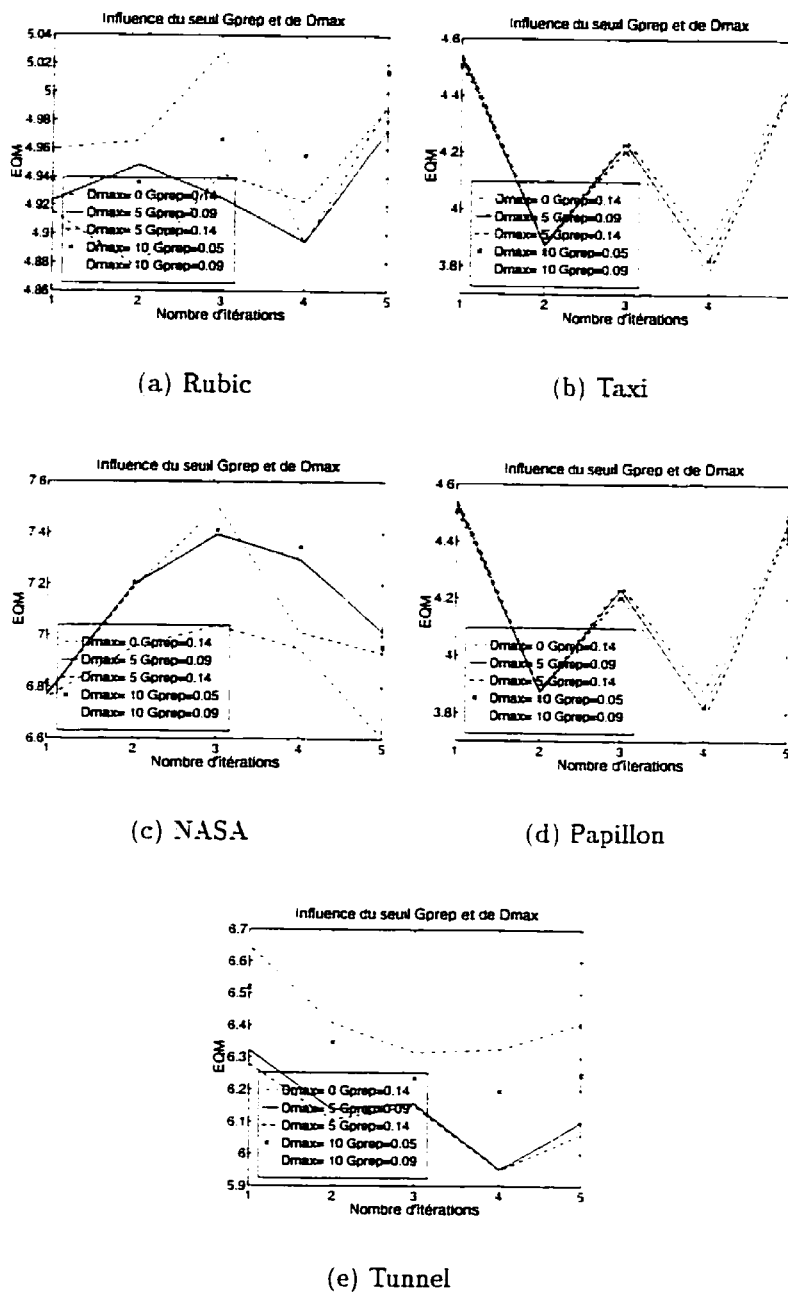


Figure D.12 : Influence du nombre maximum de termes prépondérants Δ_{prep}^{max} et du seuil de prépondérance Γ_{prep} sur l'erreur quadratique moyenne

D.3 Module de codage stéréoscopique

Les figures D.13 et D.14 illustrent l'influence du seuil de succès du module Γ_{disp} sur le taux de compression et l'erreur quadratique moyenne.

Les figures D.15 et D.16 illustrent l'influence de la longueur du domaine de recherche horizontal $\lambda_{disp,x}$ sur le taux de compression et l'erreur quadratique moyenne.

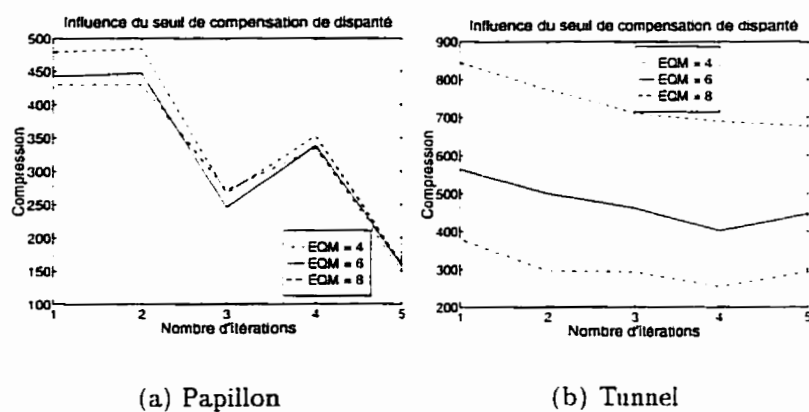


Figure D.13 : Influence du seuil de succès du module de compensation de disparité Γ_{disp} sur le taux de compression

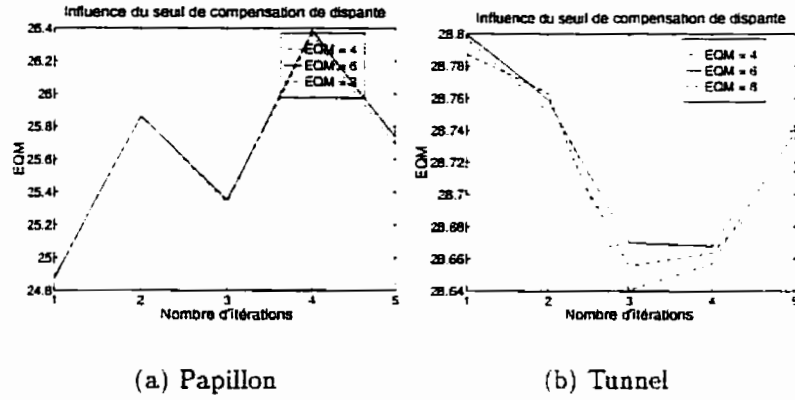


Figure D.14 : Influence du seuil de succès du module de compensation de disparité Γ_{disp} sur l'erreur quadratique moyenne

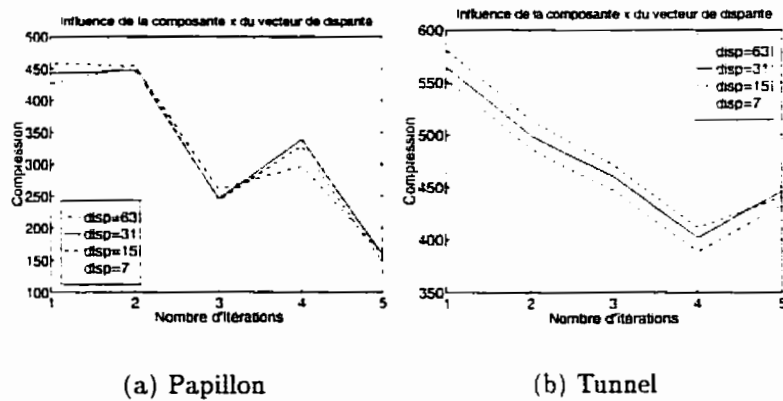


Figure D.15 : Influence de la longueur du domaine de recherche horizontal $\lambda_{disp,x}$ sur le taux de compression

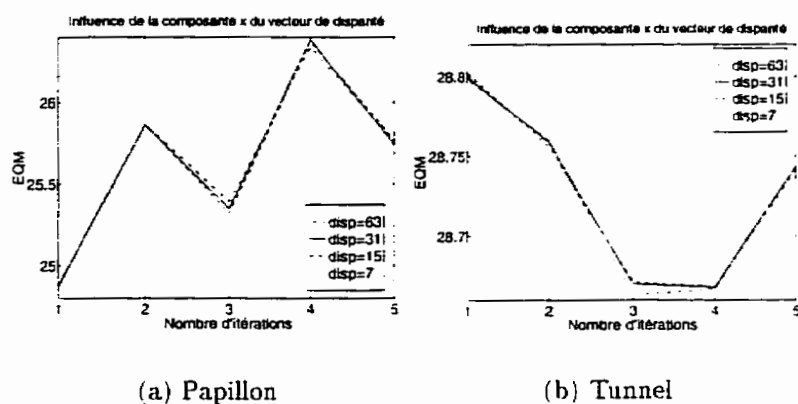
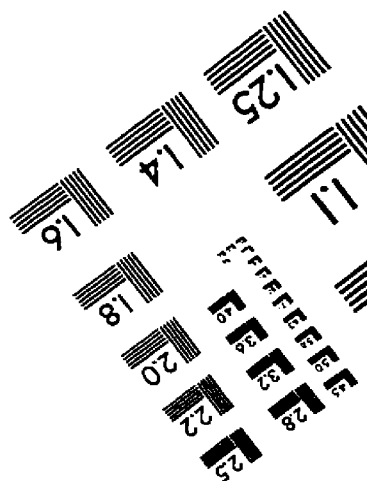
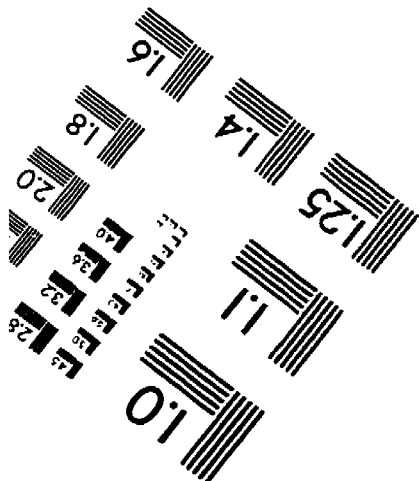
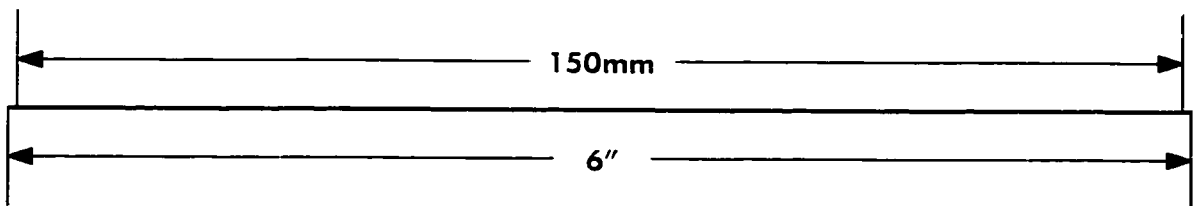
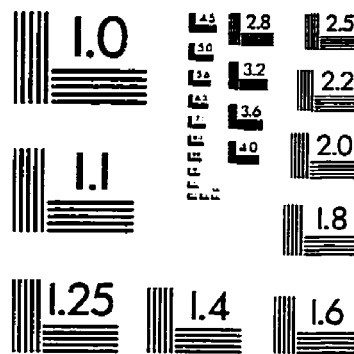
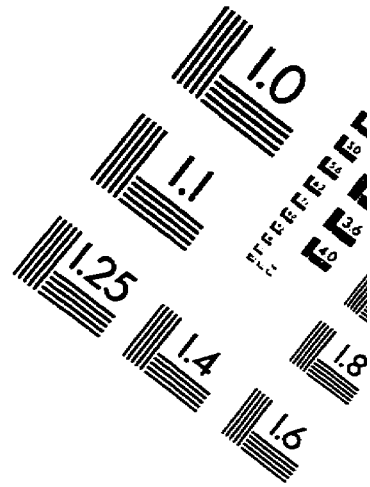
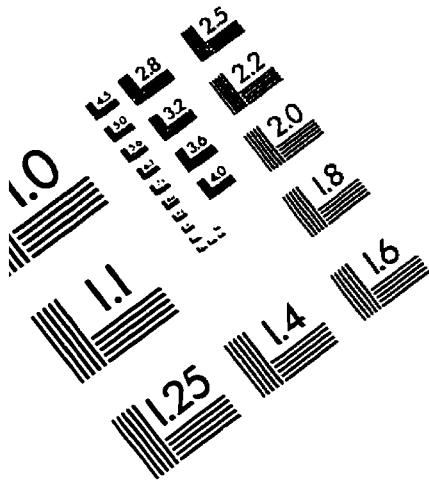


Figure D.16 : Influence de la longueur du domaine de recherche horizontal $\lambda_{disp,x}$ sur l'erreur quadratique moyenne

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc.
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved