

Dédicace

A ma famille,
A mon mari,
Et à mes trois enfants.

M^{me} Amina Belalia

Remerciements

En tout premier lieu, je remercie le bon Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace et le courage pour dépasser toutes les difficultés.

Je voudrais remercier les membres de jury, Pr. Kamel Mohamed FARAOUN, Pr. Abdelhafid BESSAID et Pr. Joseph RONSIN d'avoir accepté de relire cette thèse et d'en être examinateurs. La version finale de ce mémoire a bénéficié de leur lecture très attentive et de leurs remarques précieuses. Je tiens à remercier Pr. Nasreddine TALEB d'avoir accepté d'être président du jury. Je remercie également Pr. Shiping ZHU d'avoir accepté de co-diriger cette thèse et de la relire. Je remercie également tous les membres du jury d'avoir accepté d'assister à la présentation de ce travail, particulièrement Pr. Joseph RONSIN et Pr. Kidiyo KPALMA qui se sont déplacés depuis Rennes, France.

Mes chaleureux remerciements vont au directeur de ma thèse, professeur Kamel Belloulata, pour m'avoir guidé, encouragé, conseillé et avoir consacré une grande partie de son temps au suivi de cette thèse. Je lui remercie pour l'aide qu'il m'a apportée, et pour tous les articles qu'il m'a fournis pour mener à bien cette étude. Son écoute et ces conseils avisés ont été précieux pour l'aboutissement à ces travaux. Je lui exprime toute ma gratitude.

Je suis très reconnaissante à ma mère qui n'a cessé de m'encourager et de prier pour moi, à mes sœurs Malika et Nadia pour leur soutien moral, et à mes frères Tahar, Mohamed et Hafid pour leurs encouragements. Je n'oublie pas mes tantes et ma grande-mère pour leurs prières pour moi. Je tiens à remercier ma belle famille pour leurs encouragements de près ou de loin.

Je remercie mes enfants, Sara, Younès et Houda, pour la joie qu'ils m'apportent, ils sont aussi la source de ma motivation, j'apprends beaucoup avec eux, ils sont aujourd'hui ma fierté et mon avenir.

Je remercie spécialement celui qui partage ma vie mon époux, Kamel, qui a toujours cru en moi, pour sa sollicitude, ses encouragements qui m'ont été précieux dans les moments de doute et pour toute la patience dont il a fait preuve.

TABLE DES MATIÈRES

TABLE DES MATIÈRES	1
TABLE DES FIGURES	4
LISTE DES TABLEAUX	12
INTRODUCTION GÉNÉRALE	15
I Compression Fractale Basée-Région des Séquences Vidéo Monoculaires et Stéréoscopiques Utilisant Shape-Adaptive DCT (SA-DCT)	17
1 INTRODUCTION	19
1.1 CONTEXTE	19
1.2 PROBLÉMATIQUE ET OBJECTIFS	20
1.3 CONTRIBUTION	21
1.4 STRUCTURE DE LA DISSERTATION	21
2 CODAGE DES SÉQUENCES D'IMAGES PAR FRACTALS	23
2.1 CODAGE D'IMAGES	24
2.2 CODAGE FRACTALE DES IMAGES	27
2.2.1 Introduction à la théorie des fractales	27
2.2.2 Principe des IFS	27
2.2.3 Approche de JACQUIN	28
2.2.4 Paramètres additionnels du codec d'image	30
2.2.5 Décodage de JACQUIN	37
2.3 CODAGE DES SÉQUENCES VIDÉO	38
2.3.1 Codage Vidéo	38
2.3.2 Estimation de mouvement par la méthode des fractales	39
2.3.3 Approches pour le codage vidéo	40
2.3.4 Structure du codec fractal des séquences vidéo dans le domaine spatial	43
2.3.5 Structure du codec fractal basé-région des séquences vidéo dans le domaine spatial	45
2.4 CONCLUSION	50
3 CODAGE FRACTAL BASÉ-RÉGION DES SÉQUENCES VIDÉO UTILISANT LA SA-DCT	51
3.1 INTRODUCTION	52
3.2 RE-FORMULATION DE LA THÉORIE DES IFS DANS LE DOMAINE TRANSFORMÉ	54
3.3 CODAGE FRACTAL BASÉ-RÉGION DES IMAGES DANS LE DOMAINE TRANSFORMÉ	55

3.3.1	Approche Proposée	55
3.3.2	Choix de l'espace de recherche	58
3.3.3	Encodage des paramètres	59
3.4	CODAGE FRACTAL BASÉ-RÉGION DES SÉQUENCES VIDÉO	60
3.5	CODAGE FRACTAL BASÉ-RÉGION DES SÉQUENCES STÉRÉO	62
3.5.1	Codage basé-bloc et le profile multi-vue de MPEG2	62
3.5.2	Nouvelle méthode de codage fractal des séquences stéréo	63
3.6	RÉSULTATS EXPÉRIMENTAUX	65
3.6.1	Codage fractal basé-région de séquence vidéo	65
3.6.2	Codage fractal basé-région de séquence stéréo	77
4	CONCLUSION	79
 II Indexation Basée-Région d'Image Utilisant Shape-Adaptive DCT (SA-DCT)		81
1	INTRODUCTION	83
1.1	CONTEXTE	83
1.2	PROBLÉMATIQUE ET OBJECTIFS	83
1.3	CONTRIBUTION	84
1.4	STRUCTURE DE LA DISSERTATION	85
2	SYSTÈME D'INDEXATION ET DE RECHERCHE D'IMAGE PAR LE CONTENU UTILISANT LA DCT	87
2.1	INTRODUCTION	87
2.2	PRINCIPE DE L'INDEXATION D'IMAGE PAR LE CONTENU	88
2.3	DESCRIPTEURS D'IMAGES DANS LE DOMAINE DCT	90
2.3.1	Histogramme des AC	90
2.3.2	Histogramme des DC	90
2.3.3	Descripteur de caractéristique	92
2.4	MESURE DE SIMILARITÉ	93
2.4.1	Mesure de similarité coefficient à coefficient	93
2.4.2	Mesure de similarité croisée entre coefficients	94
2.5	ÉVALUATION DE PERFORMANCES	95
2.5.1	Précision et rappel	95
2.5.2	MAP (Mean Average Precision)	95
2.5.3	Taux moyen de récupération (Average Retrieval Rate, ARR)	96
2.5.4	Taux d'erreur égale (Equal Error Rate,EER)	96
2.6	CONCLUSION	97
3	SYSTÈME D'INDEXATION ET DE RECHERCHE D'IMAGE BASÉ-RÉGION UTILISANT LA SA-DCT	99
3.1	INTRODUCTION	99
3.2	INDEXATION D'IMAGE BASÉE-RÉGION	100
3.3	PRINCIPE DE L'APPROCHE PROPOSÉE	103
3.3.1	Indexation d'image basée-objet utilisant Shape-Adaptive DCT	103

3.3.2	Segmentation d'image	106
3.3.3	Transformation adaptée aux objets SA-DCT	107
3.3.4	Pré-traitement	107
3.3.5	Mesure de similarité	109
3.3.6	Résumé	111
3.4	RÉSULTATS EXPÉRIMENTAUX	112
3.4.1	Coût de calculs	113
3.4.2	Ajustement de paramètres	113
3.4.3	Optimisation	119
3.4.4	Système avec paramètres globaux (Fig.3.12)	122
3.4.5	Optimisation de la mesure de similarité	134
3.4.6	Type de recherche	136
3.5	CONCLUSION	139
4	CONCLUSION	141
	CONCLUSION GÉNÉRALE ET PERSPECTIVES	145
	LISTES DES PAPIERS	167
	BIBLIOGRAPHIE	169

TABLE DES FIGURES

Compression Fractale Basée-Région des Séquences Vidéo Monoculaires et Stéréoscopiques Utilisant Shape-Adaptive DCT (SA-DCT) 19

CODAGE DES SÉQUENCES D'IMAGES PAR FRACTALS	23
Figure 2.1 Chaîne de traitement comportant le codeur et le décodeur.	24
Figure 2.2 Exemples de fractales	27
Figure 2.3 Exemple d'un SFI	27
Figure 2.4 Partitionnement en bloc source et bloc cibles [16]	28
Figure 2.5 Contraction photométrique sur le bloc source D_i	30
Figure 2.6 Partitionnement en quadrees	31
Figure 2.7 Niveaux de partitionnement quadtree	31
Figure 2.8 Les trois formes canoniques selon la moyenne.	36
Figure 2.9 Les trois formes canoniques renversées	36
Figure 2.10 Décodage de l'image de Lena [16]	37
Figure 2.11 Composantes d'une séquence vidéo	38
Figure 2.12 Estimation de mouvement par la méthode des fractales	39
Figure 2.13 Codage CPM pour M trames consécutives	40
Figure 2.14 Recherche locale : (a) Spirale, (b) Balayage X-Y	41
Figure 2.15 Décodage CPM	41
Figure 2.16 Structure hybride du CPM et NCIM [51]	42
Figure 2.17 Codec hybride CPM/NCIM : (a)Codeur CPM/NCIM, (b) Décodeur CPM/NCIM	43
Figure 2.18 Changement de scène	44
Figure 2.19 Deux trames consécutives de séquence Children 10HZ	46
Figure 2.20 Deux trames consécutives d'un alpha-plane Children	47
Figure 2.21 Illustration schématique du modèle du codage fractal de la vidéo basé-région. (a) Illustration du partitionnement en quadtree pour les blocs intérieurs et les blocs frontières, le même partitionnement est utilisé pour les blocs sources et cibles. (b) Des blocs cibles Intérieurs sont associés aux blocs sources intérieurs de la même région tandis que les blocs frontières sont partiellement correspondus aux blocs sources de la même-région.	47
Figure 2.22 Exemple d'application du SA-spatial	48
Figure 2.23 Processus de codage d'un bloc cible (Extrapolation et filtrage)	49

CODAGE FRACTAL BASÉ-RÉGION DES SÉQUENCES VIDÉO EN UTILISANT SA-DCT 51

Figure 3.1	Illustration de la SA-DCT : (a) région de forme arbitraire ; (b) alignement vertical effectué par la DCT vertical 1-D ; (c) alignement horizontal effectué par la DCT horizontal 1-D.	56
Figure 3.2	Illustration schématique du codage fractal basé-région dans le domaine de la DCT du système proposé. (a) Les blocs cibles intérieurs sont associés aux blocs sources intérieurs de la même région tandis que les blocs frontières cibles sont partiellement correspondus (éq.3.2) aux blocs frontières sources de la même région. (b) Les segments de blocs frontières sont traités indépendamment : le segment $\mathcal{S}_{r_i}^1$ est calculé à partir d'un segment du bloc source d_j par une transformation affine w_i^1 , Par contre le segment $\mathcal{S}_{r_i}^2$ du même bloc cible r_i est calculé à partir d'un autre bloc source par une transformation différente. Les blocs intérieurs sont transformés par la DCT, tandis que les segments des blocs frontières sont transformés par l'utilisation de la SA-DCT	56
Figure 3.3	Exemple d'application de la SA-DCT pour un bloc frontières avec deux segments. Notons que dû aux décalages horizontaux et verticaux, la forme de \mathcal{P}_i est différente de celle de \mathcal{S}_i	57
Figure 3.4	illustration schématique de la prédiction inter-vue. (a) Le profil multi-vue de MPEG-2 [7] (b) Schéma hybride de codage de séquence stéréo en combinant CPM / NCIM avec QDE / QDC.	62
Figure 3.5	illustration schématique de la valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne pour une séquence de 20 trames(BW). (a) Séquence News-cif(b) Séquence Weather-cif.	67
Figure 3.6	illustration schématique de la valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne pour une séquence de 20 trames(BW). (a) Séquence Children-Cif(b) Séquence Cyclamen-Sif.	67
Figure 3.7	Comparaison entre le codage fractal basé-région (BR-Plein) et le codage fractal non basé-région(NBR) dans le domaine spatial pour la séquence News de 20 trames.	68
Figure 3.8	Comparaison entre le codage fractal basé-région (BR-Plein) et le codage fractal non basé-région(NBR) dans le domaine spatial pour la séquence Children de 20 trames.	68
Figure 3.9	Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence News de 20 trames.	69
Figure 3.10	Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence Children de 20 trames.	69
Figure 3.11	Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence Weather de 20 trames.	70
Figure 3.12	Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence Cyclamen de 20 trames.	70

Figure 3.13	Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence <i>News</i> de 20 trames. (a) La valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne (BW), (b) Temps de codage en fonction de la largeur de bande moyenne (BW).	71
Figure 3.14	Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence <i>Children</i> de 20 trames. (a) La valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne (BW), (b) Temps de codage en fonction de la largeur de bande moyenne (BW).	72
Figure 3.15	Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence <i>Weather</i> de 20 trames. (a) La valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne (BW), (b) Temps de codage en fonction de la largeur de bande moyenne (BW).	72
Figure 3.16	Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence <i>Cyclamen</i> de 20 trames. (a) La valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne (BW), (b) Temps de codage en fonction de la largeur de bande moyenne (BW).	73
Figure 3.17	Comparaison entre le codage fractal basé-région dans le domaine fréquentiel et le codage fractal basé-région dans le domaine spatial pour la séquence <i>News</i> de 20 trames.	73
Figure 3.18	Comparaison entre le codage fractal basé-région dans le domaine fréquentiel et le codage fractal basé-région dans le domaine spatial pour la séquence <i>Children</i> de 20 trames.	74
Figure 3.19	Comparaison entre le codage fractal basé-région dans le domaine fréquentiel et le codage fractal basé-région dans le domaine spatial pour la séquence <i>Weather</i> de 20 trames.	74
Figure 3.20	Comparaison entre le codage fractal basé-région dans le domaine fréquentiel et le codage fractal basé-région dans le domaine spatial pour la séquence <i>Cyclamen</i> de 20 trames.	74
Figure 3.21	Comparaison subjective entre (a) le codage fractal basé-région dans le domaine fréquentiel, (b) le codage fractal basé-région dans le domaine spatial et (c) le codage fractal non basé-région dans le domaine fréquentiel, pour la trame N^o 5 de la séquence <i>Cyclamen</i>	75
Figure 3.22	Trame N^o 5 décodée de la séquence <i>Cyclamen</i> . (a) Premier-plan du codage fractal basé-région dans le domaine fréquentiel, (b) Premier-plan du codage fractal non basé-région dans le domaine fréquentiel.	75
Figure 3.23	La cinquième trame décodée de la séquence stéréo <i>Anne</i> (a) Premier-plan <i>droit</i> (b) Premier-plan <i>gauche</i> (c) trame complète <i>droite</i> (d) Trame complète <i>gauche</i>	78
Figure 3.24	La cinquième trame décodée de la séquence stéréo <i>Sergio</i> (a) Premier-plan <i>droit</i> (b) Premier-plan <i>gauche</i> (c) trame complète <i>droite</i> (d) Trame complète <i>gauche</i>	78

Indexation Basée-Région d'Image Utilisant Shape-Adaptive DCT (SA-DCT)	83
SYSTÈME D'INDEXATION ET DE RECHERCHE D'IMAGE PAR LE CONTENU UTILISANT LA DCT	87
Figure 2.1 Architecture générale d'un système d'indexation et de recherche d'image par le contenu (CBIR).	89
Figure 2.2 <i>Processus de construction du AC-pattern [9]</i>	91
Figure 2.3 <i>Histogramme des 100 AC-patterns, hautes fréquences des occurrences de l'image African (CBIR)</i>	91
Figure 2.4 Histogramme des 50 AC-patterns, hautes fréquences des occurrences de l'image (CBIR) (Content-Based Image Retrieval).	91
Figure 2.5 Processus de construction du DC-pattern [86]	92
Figure 2.6 Histogramme combiné des premières occurrences de hautes fréquences du AC-patterns et DC-patterns pour le CBIR.	93
SYSTÈME D'INDEXATION ET DE RECHERCHE D'IMAGE BASÉ-RÉGION UTILISANT LA SA-DCT	99
Figure 3.1 Bloc avec deux segments S^1 et S^2	103
Figure 3.2 <i>Schéma synoptique de la méthode basée-région proposée</i>	104
Figure 3.3 <i>Processus de construction de l'histogramme combiné par la méthode proposée. (a) Histogramme combiné en tenant compte des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$. (b) Illustration schématique montrant une image composée de deux objets, premier-plan et arrière-plan. Pour chaque objet, les blocs intérieurs sont transformés par la DCT, tandis que les segments des blocs frontières sont transformés par l'utilisation de la SA-DCT. H_{Fore} est construit à partir des blocs intérieurs et des segments des blocs frontières du premier-plan, alors que H_{Back} est construit à partir des blocs intérieurs et des segments des blocs frontières de l'arrière-plan. Une concaténation générale est définie en appliquant β pour combiner l'histogramme premier-plan H_{Fore} avec l'histogramme arrière-plan H_{Back}.</i>	105
Figure 3.4 Histogramme des 50 AC-patterns, hautes fréquences des occurrences de l'image (RBIR) (Region-Based Image Retrieval).	105
Figure 3.5 Histogramme combiné des premières occurrences de hautes fréquences du AC-pattern et DC-pattern pour le RBIR (Region-Based Image Retrieval).	106
Figure 3.6 <i>Exemple de calcul de similarité entre les images</i>	110
Figure 3.7 <i>Schéma blocs de la méthode RBIR proposée utilisant la transformation SA-DCT</i>	111
Figure 3.8 <i>Processus d'apprentissage optimisant les paramètres d'indexation $QP_{AC}, QP_{DC}, \gamma, NC, ACBins, DCBins$.</i>	114
Figure 3.9 <i>Processus d'apprentissage optimisant les paramètres de la recherche $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$.</i>	116
Figure 3.10 MAP en fonction de la variation du paramètre β pour la base de données Corel-1000	117
Figure 3.11 MAP en fonction de la variation du paramètre β pour chaque classe : les courbes (1)-(10) sont pour les classes de la base de données Corel-1000	118

Figure 3.12	schéma d'illustration montrant deux Systèmes résultants par la variation des paramètres : <i>Système avec paramètres globaux</i> et <i>Système avec paramètres adaptés pour chaque classe</i>	119
Figure 3.13	<i>Processus de construction de l'histogramme combiné par la méthode proposée : principe de l'optimisation de la recherche. (a) Histogramme combiné en tenant compte des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ et $\{ \gamma_{CF}, \gamma_{PF}, \gamma_{CB}, \gamma_{PB} \}$. (b) Schéma d'illustration montrant une image composée de deux objets, premier-plan et arrière-plan. L'histogramme du premier-plan, H_{Fore}, est combiné à partir de l'histogramme issu des blocs intérieurs, H_{PF}, et l'histogramme issu des segments de blocs frontières, H_{CF}, du premier-plan. Alors que l'histogramme de l'arrière-plan, H_{Back}, est combiné à partir de l'histogramme issu des blocs intérieurs, H_{PB}, et l'histogramme issu des segments de blocs frontières, H_{CB}, de l'arrière-plan. L'histogramme combiné global est calculé à partir des histogrammes premier-plan, H_{Fore}, et arrière-plan, H_{Back}.</i>	119
Figure 3.14	Histogramme combiné des premières occurrences de hautes fréquences du AC-Patterns et DC-Patterns pour le RBIR (Region-Based Image Retrieval) : Principe de l'optimisation.	121
Figure 3.15	Les courbes précision-rappel moyennes entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement). Les catégories d'images (a)"Fleurs" (b)"Montagnes" sont de la base de données Corel-1000, et les catégories d'images (c)"Gants de baseball" (d)"Sacs-à-dos" sont de la base de données Caltech-256	123
Figure 3.16	Les courbes précision-rappel moyennes : (a)Base de données Corel-1000, (b) Base de données Caltech-256, entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement)[86]	124
Figure 3.17	Les courbes précision-rappel moyennes entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement). Les catégories d'images (a)"People 33" (b)"People 50" sont de la base de données GTF	125
Figure 3.18	Comparaison des résultats pour la base de données GTF : a) EER b) Les courbes précision-rappel moyennes.	125
Figure 3.19	Optimisation : Les courbes précision-rappel moyennes entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement). Les catégories d'images (a)"Fleurs" et (b)"Montagnes" sont de la base de données Corel-1000, et les catégories d'images (c)"Gants de baseball" et (d)"Sacs-à-dos" sont de la base de données Caltech-256	126
Figure 3.20	Optimisation : Les courbes précision-rappel moyennes : (a)Base de données Corel-1000, (b) Base de données Caltech-256, entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement)[86]126	
Figure 3.21	Optimisation : Les courbes précision-rappel moyennes entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement). Les catégories d'images (a)"People 33" (b)"People 50" sont de la base de données GTF	127
Figure 3.22	Optimisation : Comparaison des résultats pour la base de données GTF : Les courbes précision-rappel moyennes.	127

Figure 3.23	Mean Average Precision (MAP) (a) Base de données Corel-1000, (b) Base de données Caltech-256	129
Figure 3.24	Robustesse de notre <i>Système avec paramètres globaux</i> pour les variations d'images pour la base de données Corel-1000	131
Figure 3.25	Robustesse du <i>Système avec paramètres globaux</i> pour les variations d'images pour la base de données Caltech-256	131
Figure 3.26	Quelques exemples d'évaluation de robustesse sur la base de données Corel-1000. La première image est l'image requête, ensuite les 5 premières images recherchées par l'utilisation du <i>Système avec paramètres globaux</i> proposé.	132
Figure 3.27	<i>Optimisation de la mesure de similarité : exemple de calcul de similarité entre les images.</i>	134
Figure 3.28	<i>Les courbes précision-rappel pour la base de données Corel-1000 calculées par la mesure de similarité image par image. (a) Chaque région de l'image requête est comparée avec toutes les régions de la base de données (similaires et non-similaires). (b) Chaque région de l'image requête est comparée avec seulement les régions similaires de la base de données.</i>	135
Figure 3.29	La moyenne P(N) (N = 10 : 10 : 100) (%) de la recherche région-par-région et image-par-image. (a) Plage, (b) Bâtiment, de la base de donnée Corel-1000	136
Figure 3.30	Réduction de l'influence négative des régions en interférences. (a) La première image est l'image requête, les sept autres images sont le résultat RBIR du système proposé basé sur la mesure de similarité image par image, (b) La première image est l'image requête, les sept autres images sont le résultat RBIR du système proposé basé sur la mesure de similarité région par région	137
Figure 3.31	La moyenne P(N) (N = 10 : 10 : 100) (%) de la recherche région-par-région et image-par-image. (a) Africain, (b) Plat, de la base de donnée Corel-1000	138
Figure 3.32	Diminution de l'influence négative de la perte importante d'information. (a) La première image est l'image requête , les sept autres images sont le résultat RBIR du système proposé basé sur la mesure de similarité région par région, (b) La première image est l'image requête, les sept autres images sont le résultat RBIR du système proposé basé sur la mesure de similarité image par image	138

Glossaire et Annexes **149**

ANNEXE A **151**

Figure 1	Échantillon d'images et leurs masques pour chaque classe de la base de données Corel-1000.	151
Figure 2	Échantillon d'images et leurs masques pour chaque classe de la base de données Caltech-256.	151
Figure 3	Échantillon d'images et leurs masques pour quelques classes de la base de données GTF.	151

ANNEXE B **153**

Figure 1	Les courbes précision-rappel moyennes entre l'approche basée-objet proposée (DCT+SA-DCT) et l'approche conventionnelle basée sur le contenu (CBIR avec DCT seulement). Les catégories d'images (a)"Fleurs" (b)"Montagnes" sont de la base de données Corel-1000, et les catégories d'images (c)"Gants de baseball" (d)"Sacs-à-dos" sont de la base de données Caltech-256	154
Figure 2	Les courbes précision-rappel moyennes : (a)Base de données Corel-1000, (b) Base de données Caltech-256, entre l'approche proposée basée-objet (DCT+SA-DCT) et l'approche conventionnelle basée sur le contenu (CBIR avec DCT seulement)[86]	154
Figure 3	Comparaison des résultats pour la base de données GTF : a) People 33 b) People 50 c) Les courbes précision-rappel moyennes.	155
Figure 4	Optimisation : les courbes précision-rappel moyennes entre l'approche proposée basée-région (DCT+SA-DCT) et l'approche conventionnelle basée sur le contenu (CBIR avec DCT seulement). Les catégories d'images (a)"Fleurs" (b)"Montagnes" sont de la base de données Corel-1000, et les catégories d'images (c)"Gants de baseball" (d)"Sacs-à-dos" sont de la base de données Caltech-256	156
Figure 5	Optimisation : les courbes précision-rappel moyennes : (a)Base de données Corel-1000, (b) Base de données Caltech-256, entre l'approche proposée basée-objet (DCT+SA-DCT) et l'approche conventionnelle basée sur le contenu (CBIR avec DCT seulement)[86]	157
Figure 6	Optimisation : comparaison des résultats pour la base de données GTF : a) People 33 b) People 50 c) Les courbes précision-rappel moyennes.	157
Figure 7	Mean Average Precision (MAP) pour la Base de données Corel-1000	158
Figure 8	Robustesse du <i>Système avec paramètres adaptés pour chaque classe</i> pour les variations d'images pour la base de données Corel-1000	160
Figure 9	Robustesse du <i>Système avec paramètres adaptés pour chaque classe</i> pour les variations d'images pour la base de données Caltech-256	160
Figure 10	Quelques exemples d'évaluation de la robustesse sur la base de données Corel-1000. La première image est l'image requête, ensuite les 5 premières images recherchées par l'utilisation du <i>Système avec paramètres adaptés pour chaque classe</i> proposée.	161
ANNEXE C		163
Figure 1	Interface graphique de la méthode proposée.	163
Figure 2	Exemple de recherche par le <i>système classique (CBIR avec DCT seulement)</i> sur la base de données Corel-1000 : (a) Image montagne. (b) Image Cheval. La première image est l'image requête, les 19 autres images sont les images recherchées par le <i>système classique (CBIR avec DCT seulement)</i>	164
Figure 3	Exemple de recherche par le <i>système classique (CBIR avec DCT seulement)</i> sur la base de données Caltech-256 : (a) Image AK-47. (b) Image Drapeau Américain. La première image est l'image requête, les 19 autres images sont les images recherchées par le <i>système classique (CBIR avec DCT seulement)</i> proposée.	164

Figure 4	Exemple de recherche par le <i>système classique (CBIR avec DCT seulement)</i> sur la base de données GTF : (a) Image Classe 1. (b) Image Classe 10. La première image est l'image requête, les 14 autres images sont les images recherchées par le <i>système classique (CBIR avec DCT seulement)</i> proposée. 164
Figure 5	Exemple de recherche par le <i>système avec paramètres globaux</i> sur la base de données Corel-1000 : (a) Image montagne. (b) Image Cheval. La première image est l'image requête, les 19 autres images sont les images recherchées par le <i>système avec paramètres globaux</i> proposée. 165
Figure 6	Exemple de recherche par le <i>système avec paramètres globaux</i> sur la base de données Caltech-256 : (a) Image AK-47. (b) Image Drapeau Américain. La première image est l'image requête, les 19 autres images sont les images recherchées par le <i>système avec paramètres globaux</i> proposée. 165
Figure 7	Exemple de recherche par le <i>système avec paramètres globaux</i> sur la base de données GTF : (a) Image Classe 1. (b) Image Classe 10. La première image est l'image requête, les 14 autres images sont les images recherchées par le <i>système avec paramètres globaux</i> proposée. 165
Figure 8	Exemple de recherche par le <i>système avec paramètres adaptés pour chaque classe</i> sur la base de données Corel-1000 : (a) Image montagne. (b) Image Cheval. La première image est l'image requête, les 19 autres images sont les images recherchées par le <i>système avec paramètres adaptés pour chaque classe</i> proposée. . 166
Figure 9	Exemple de recherche par le <i>système avec paramètres adaptés pour chaque classe</i> sur la base de données Caltech-256 : (a) Image AK-47. (b) Image Drapeau Américain. La première image est l'image requête, les 19 autres images sont les images recherchées par le <i>système avec paramètres adaptés pour chaque classe</i> proposée. 166
Figure 10	Exemple de recherche par le <i>système avec paramètres adaptés pour chaque classe</i> sur la base de données GTF : (a) Image Classe 1. (b) Image Classe 10. La première image est l'image requête, les 14 autres images sont les images recherchées par le <i>système avec paramètres adaptés pour chaque classe</i> proposée. . 166

Liste des tableaux

Compression Fractale Basée-Région des Séquences Vidéo Monoculaires et Stéréoscopiques Utilisant Shape-Adaptive DCT (SA-DCT) 19

CODAGE DES SÉQUENCES D'IMAGES PAR FRACTALS	23
Table 2.1 Organigramme du partitionnement en quadtree selon un seuil de tolérance . .	32
Table 2.2 Organigramme du partitionnement en quadtree selon un nombre total de transformations.	32
Table 2.3 Influence de la zone de recherche	33

CODAGE FRACTAL BASÉ-RÉGION DES SÉQUENCES VIDÉO EN UTILISANT SA-DCT 51

Table 3.1 Paramètres fractals et d'implémentations pour le système proposé	65
Table 3.2 Effet de Min-part et Max-part du partitionnement quadtree sur le codage basé-région de la séquence News-Cif codée en 20 trames.	66
Table 3.3 Effet de Min-part et Max-part du partitionnement quadtree sur le codage basé-région de la séquence Weather-Cif codée en 20 trames.	66
Table 3.4 Résultats du codage sur la séquence stéréo <i>Anne</i>	77
Table 3.5 Résultats du codage sur la séquence stéréo <i>Sergio</i>	77

Indexation Basée-Région d'Image Utilisant Shape-Adaptive DCT (SA-DCT) 83

SYSTÈME D'INDEXATION ET DE RECHERCHE D'IMAGE BASÉ-RÉGION UTILISANT LA SA-DCT 99

Table 3.1 Similarité entre les régions	110
Table 3.2 Les catégories de Corel-1000 des images testées	112
Table 3.3 Les catégories de Caltech-256 des images testées	112
Table 3.4 H_{Full} pour les combinaisons binaires des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$	116
Table 3.5 Valeurs optimales de $\{ \beta, \alpha_{Fore}, \alpha_{Back}, \alpha(DCT) \}$ pour la base de données Corel-1000	122
Table 3.6 Valeurs optimales de $\{ \beta, \alpha_{Fore}, \alpha_{Back}, \alpha(DCT) \}$ pour la base de données Caltech-256	122
Table 3.7 Valeurs optimales de $\{ \beta, \alpha_{Fore}, \alpha_{Back}, \alpha(DCT) \}$ pour la base de données GTF	124
Table 3.8 Précision Moyenne(AP) des différentes méthodes pour la base de données Corel-1000	128

Table 3.9	Précision Moyenne(AP) des différentes méthodes pour la base de données Caltech-256	128
-----------	--	-----

Glossaire et Annexes **149**

ANNEXE B **153**

Table 1	$\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ pour chaque classe pour la base de données Corel-1000 . . .	153
Table 2	$\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ pour chaque classe pour la base de données Caltech-256 . . .	153
Table 3	Précision Moyenne(AP) des différentes méthodes pour la base de données Corel-1000	158
Table 4	Précision Moyenne(AP) des différentes méthodes pour la base de données Caltech-256	158

INTRODUCTION GÉNÉRALE

*Trop de connaissance ne facilite pas
les plus simples décisions.*

*Les enfants de Dune
Frank Herbert (1920-1986).*

Dans cette thèse, deux systèmes basés-régions et basés sur la transformation SA-DCT (Shape Adaptive DCT) sont présentés. Le premier système est un système de compression des séquences vidéo monoculaires et stéréoscopiques, alors que le deuxième système est un système d'indexation et de recherche d'images.

L'objectif du travail proposé est la réalisation de ces deux systèmes puis la confirmation de leur efficacité et de leurs justesses. Le mémoire comporte deux parties :

La partie (I) détaillera le système de compression des séquences vidéo monoculaires et stéréoscopiques proposé, notre travail a pour objectif de contribuer au codage basé-région de séquences vidéo et stéréo en utilisant les fractales et le shape adaptive DCT. La méthode proposée offre comme bénéfice de bien coder les frontières entre deux régions (objets), offrant une amélioration nette de la qualité (PSNR) à la sortie du décodeur. On peut aussi bénéficier de la corrélation temporelle élevée entre deux trames consécutives pour réduire le débit binaire de transmission et réduire le temps d'encodage des séquences. La partie (II) proposera le système d'indexation et de recherche d'images, notre travail propose une nouvelle approche dans le domaine de la recherche d'images par région (RBIR), utilisant la transformation cosinus discrète adaptée aux objets (SA-DCT). Dans ce système de recherche, une image est représentée par des régions segmentées, et chaque région est associée à un vecteur de caractéristiques dérivés de la DCT et la SA-DCT. Chaque région de l'image requête, peut être sélectionnée comme l'objet principal. Pour les images qui n'ont pas un objet principal, l'utilisateur pourra toujours sélectionner l'image entière comme requête. Les résultats obtenus montrent que l'approche proposée est capable d'identifier l'objet principal et de réduire l'influence de l'arrière-plan, ceci améliore la performance de la recherche d'images en comparaison avec d'autres systèmes conventionnels utilisant la DCT dans le domaine CBIR et même avec des systèmes dans le domaine RBIR.

Pour chacune des parties, des résultats expérimentaux sont présentés et bien détaillés, puis des conclusions sont tirées.

I

Compression Fractale Basée-Région des Séquences Vidéo Monoculaires et Stéréoscopiques Utilisant Shape-Adaptive DCT (SA-DCT)

INTRODUCTION



1.1 CONTEXTE

Depuis de nombreuses années, l'image numérique est un outil favorisé dans des domaines très divers telle que la médecine, l'imagerie satellitaire et la télévision numérique. Dans ces domaines, lorsqu'on parle d'images et de séquences vidéo, ces sources d'informations brutes représentent une quantité de données très importantes et contiennent beaucoup de textures redondantes. D'où la nécessité de mise en place des méthodes d'encodage et de compression, avant la transmission ou le stockage, qui permettent de représenter la même information mais avec beaucoup moins de données.

Pour ce faire, différentes méthodes basées sur la compression des données à transmettre ou à stocker ont vu le jour. Ces méthodes de compression permettent de détecter et de réduire les redondances entre les pixels de l'image originale, puis de quantifier et de coder l'information quasi-décorrélée. D'une façon générale, on peut classer ces méthodes en 2 grandes catégories :

la compression sans perte de données.

la compression avec perte de données qui permet d'obtenir d'excellents résultats mais qui entraîne inévitablement une altération des données d'origine. Plus les taux de compression sont élevés, plus cette perte est importante, mais plus l'image décodée est déformée. Ceci montre la nécessité de trouver un compromis entre le rapport débit/distorsion.

Beaucoup de standards sont développés dans le domaine de la compression des images et des vidéos tels que JPEG [76] (Joint Photographic Expert Group) pour les images fixes et MPEG [40] pour les séquences d'images. Parmi les standards existants, la norme MPEG-4 est largement utilisée sur Internet grâce au taux de compression significatif qu'elle offre. Le codec MPEG-4 effectue le codage d'objets qui nécessite une mise à jour pour compenser leurs mouvements ou leurs déformations dans une séquence vidéo. Il s'agit d'une méthode adaptative puisque les objets fixes ne sont codés qu'une seule fois, c-à-d lors de leurs apparitions dans la séquence.

1.2 PROBLÉMATIQUE ET OBJECTIFS

Le codage de la vidéo est un domaine très vaste, beaucoup de recherches sont en cours qui ont pour but d'améliorer les techniques de codage pour obtenir des résultats performants. Pour comparer entre les différentes méthodes existantes on considère quatre critères intéressants :

- Taux de compression (débit binaire).
- Taux de distorsion (qualité de codage).
- Temps de codage.
- Qualité visuels.

Pour coder les images, on utilise des méthodes de codage comme la DCT (Discrete Cosine Transform) et les ondelettes (Wavelets). Pour la vidéo, des méthodes d'estimation et de compensation de mouvement sont souvent utilisées pour considérer l'aspect temporel : on parle ainsi des deux algorithmes CPM et le NCIM.

L'objectif de notre travail est le développement d'une méthodologie optimisée de compression des séquences vidéo basées-région fondée sur la théorie des fractales dans le domaine transformé en utilisant le Shape Adaptive DCT.

La théorie des fractales était bien étudiée, en math et physique, bien avant les années 80, après elle a été introduite par M. Barnsley [12] ou [13] en compression des images. Le codage fractal a ensuite vu une croissance dans le domaine du codage d'image. Cette méthode de compression se base sur la théorie des Systèmes de Fonctions Itératives (IFS) et s'appuie sur le théorème de collage du Barnsley. Il s'agit de partitionner l'image et de mettre en correspondance chaque partie élémentaire obtenue avec une autre partie de taille supérieure recherchée dans toute l'image (l'auto-similarité).

Le codage de la vidéo est basé sur le codage d'image. La vidéo contient une composante dynamique et temporelle en plus de contenir les deux dimensions spatiales de l'image. Il s'agit de bien comprendre les propriétés de la vidéo avant d'adapter un codec d'images fixes au codage de séquences.

Le codage de la vidéo basé-région permet de coder les régions ou objets dans les images de la séquence de façon indépendante et d'offrir plus de flexibilité sur le contrôle de la largeur de bande pour la transmission. Ce codec permet de mieux coder les frontières de différentes régions.

Plusieurs algorithmes ont déjà été proposés pour effectuer ce type de codage sur une image fixe. Il existe une méthode de codage basée-région par fractales sur une image fixe en utilisant la DCT de forme adaptative (« Shape-Adaptive DCT » ou « SA-DCT »)[17],[19]. L'algorithme du SA-DCT peut être intégré au codec hybrid CPM-NCIM de

la vidéo pour faire le codage des séquences vidéo segmentées. Les objets seront ainsi séparés et codés indépendamment.

1.3 CONTRIBUTION

Notre travail a pour objectif de contribuer au codage basé-région de séquences vidéo et stéréo en utilisant les fractales et le shape adaptive DCT. La méthode proposée offre comme bénéfice de bien coder les frontières entre deux régions (objets), offrant une amélioration nette de la qualité (PSNR) à la sortie du décodeur. On peut aussi bénéficier de la corrélation temporelle élevée entre deux trames consécutives pour réduire le débit binaire de transmission et réduire le temps d'encodage des séquences.

1.4 STRUCTURE DE LA DISSERTATION

Dans le chapitre 2, on présentera brièvement la théorie des fractales et on introduira le codage des images fixes. On détaillera aussi, les différentes méthodes utilisées pour coder la vidéo, on parlera ainsi des deux algorithmes, CPM et NCIM, et on présentera le codec fractal basé-région hybride CPM-NCIM dans le domaine spatial. Nous proposerons au chapitre 3 l'algorithme du Shape Adaptive DCT et son intégration dans le codec hybride CPM-NCIM, puis nous présenterons les résultats obtenus qui seront alors analysés. Nous identifierons les améliorations et les inconvénients apportés par l'ajout de la SA-DCT au codage basé-région. Nous concluons cette étude en soulignant les résultats importants obtenus et en dégageant les principales perspectives.

CODAGE DES SÉQUENCES D'IMAGES PAR FRACTALS

2

Ce chapitre a pour objectifs de présenter les principes du codage par fractales et de faire une synthèse des méthodes de compression des séquences vidéo. Mais en premier lieu, il peut être avantageux de comprendre les concepts du codage d'une image fixe. On présentera donc les opérations de base d'un codeur d'image fixe pour passer ensuite à la théorie des fractales et le codage d'images par fractales pour développer la méthode proposée par Jacquin [44]. On détaillera par la suite le codage des séquences d'images et on parlera des méthodes proposées, parmi lesquelles le CPM et le NCIM. On décrira la structure du codec hybride CPM-NCIM. À la fin de ce chapitre, on parlera du codec hybride basé-région dans le domaine spatial.

2.1 CODAGE D'IMAGES

Le codage d'image est la représentation d'une image fixe à l'aide de mots de code dont la quantité de données est inférieure à celle de la source originale. Pour obtenir des taux de compression importants, des méthodes d'approximation sont souvent utilisées et elles induisent une distorsion ou des erreurs dans la sortie décodée. Le codage étant plus ou moins complexe, il nécessite un certain temps d'exécution avant qu'il soit complété. Bref, il faut opter pour un faible débit, une bonne qualité de décodage et un temps de codage minimal. Plusieurs travaux de recherche ont été effectués pour améliorer la qualité visuelle désirée de l'image sans augmenter le débit de transmission. En effet, certains algorithmes de traitement sont implémentés dans l'encodeur (pré-processeur) et dans le décodeur (post-processeur), affectant de peu leur performance.

La figure suivante représente un schéma bloc de l'encodeur et du décodeur d'images figure(Fig.2.1).

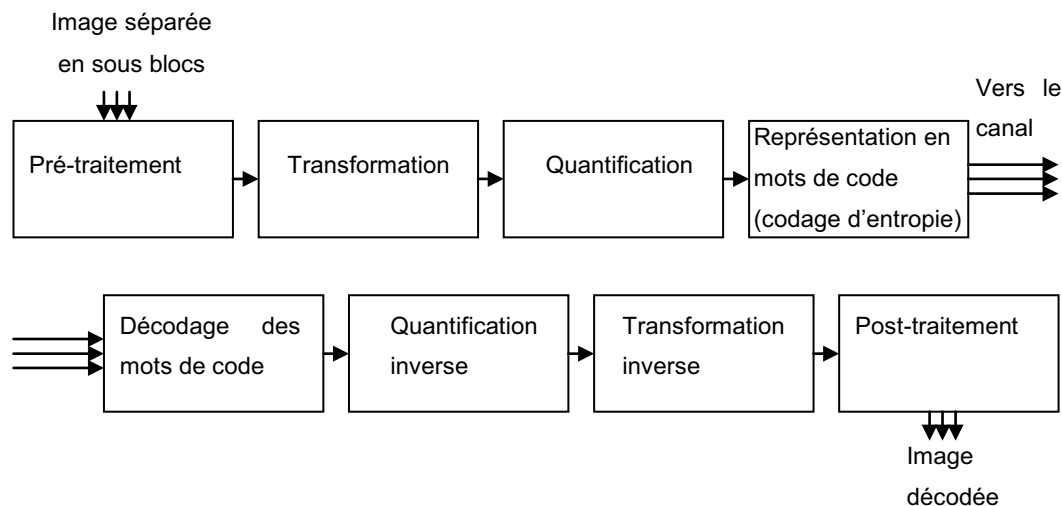


Figure 2.1 – Chaîne de traitement comportant le codeur et le décodeur.

L'encodeur d'images comporte trois opérations principales :

- Transformation.
- Quantification des coefficients.
- Représentation de l'information quantifiée en mot de code (codage entropique).

La transformation prend l'information bloc par bloc appartenant au domaine spatial et la transforme dans un autre domaine. La transformée de Fourier sert à passer au domaine fréquentiel où il peut y avoir un compactage d'énergie en quelques coefficients. Les pro-

priétés dans le domaine spectral sont différentes et peuvent s'avérer intéressantes pour la manipulation des coefficients. Par exemple, le codage de zone fixe [47] permet d'arrondir plusieurs coefficients à zéro en exploitant leurs propriétés. Il est ensuite possible de coder une suite de zéros avec un seul mot de code (Run-Length Encoding).

Dans la compression d'images, la transformation la plus utilisée est la DCT (Discrete Cosine Transform) qui a été adoptée dans les standards JPEG pour l'image fixe et MPEG pour la vidéo. On trouvera d'autres transformations plus récentes comme les ondelettes utilisées dans le standard JPEG2000. Beaucoup de littérature expliquant ces méthodes peuvent être trouvées sur Internet. Dans le chapitre 3 on détaillera le Shape Adaptive DCT, qui est la DCT adaptée aux objets.

La deuxième opération est la quantification des coefficients. La quantification sert à contrôler le passage du continu au discret au point de vue information. Donc des valeurs réelles sont converties en valeurs entières pour pouvoir les coder en mots binaires par la suite. Cette méthode d'arrondissement introduit des erreurs de quantification. La quantification est donc la seule opération responsable de la distorsion obtenue dans tout le processus du codage.

En JPEG, la quantification est l'étape dans laquelle se produit la majeure partie de la perte d'information (et donc de la qualité visuelle), mais c'est aussi celle qui permet de gagner le plus de place. La DCT a retourné, pour chaque bloc, une matrice de 8x8 nombres (dans l'hypothèse que les blocs de l'image font 8x8 pixels). La quantification consiste à diviser cette matrice par une autre, appelée matrice de quantification, et qui contient 8x8 coefficients spécifiquement choisis par le codeur. Le but ici est d'atténuer les hautes fréquences, c'est-à-dire celles auxquelles l'œil humain est très peu sensible. Ces fréquences ont des amplitudes faibles, et elles sont encore plus atténuées par la quantification ; certains coefficients sont même souvent ramenés à 0. Le calcul permettant la quantification est le suivant :

$$F^*(x, y) = \left\lfloor \frac{F(x, y) + \lfloor \frac{Q(u, v)}{2} \rfloor}{q(u, v)} \right\rfloor \cong \left(\frac{F(u, v)}{Q(u, v)} \right) \quad (2.1)$$

Avec : $\lfloor x \rfloor$ entier directement inférieur à x .

Et pour la quantification inverse :

$$\hat{F}(u, v) = F^*(u, v) \cdot Q(u, v) \quad (2.2)$$

La quantification peut être effectuée sur un seul élément d'information à la fois (quantification scalaire) ou bien sur plusieurs éléments (quantification vectorielle). La matrice (éq.2.3) contient les pas d'arrondissement pour effectuer la quantification scalaire d'un bloc de 8x8 coefficients. Cette matrice est utilisée dans le standard JPEG pour arrondir les coefficients DCT de la composante Y (luminosité) sur quelques bits.

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad (2.3)$$

L'avantage de la quantification vectorielle (QV) est que plusieurs éléments sont codés par un seul mot de code de quelques bits. La QV offre donc un meilleur taux de compression par rapport à la quantification scalaire pour un niveau semblable de distorsion.

La troisième opération est la représentation de l'information quantifiée en mots de code, plus connue sous le terme de codage entropique. Le but est de transformer la représentation d'une source de données en mots de code plus court pour sa compression ou sa transmission sur un canal de communication. La quantité d'informations à envoyer peut-être réduite sans affecter la qualité du codage. Cette partie de l'encodeur est sans perte d'information. Les méthodes d'encodage entropique les plus courantes sont le codage de Huffman, le Run-Length Encoding (RLE) et le codage arithmétique, une méthode brevetée et utilisée dans le JPEG2000.

En plus des trois opérations classiques, on retrouve des modules supplémentaires qui effectuent le pré-traitement (encodeur) ou post-traitement (décodeur). Le pré-traitement vient juste avant de passer au codage il facilite le codage et augmente la performance du codec. Le post-traitement améliore la qualité de l'image décodée donc il permet d'améliorer la qualité visuelle.

Au décodeur, on retrouve les mêmes blocs, mais avec les fonctions inverses et dans le sens inverse comme illustré à la figure (Fig.2.1).

2.2 CODAGE FRACTALE DES IMAGES

2.2.1 Introduction à la théorie des fractales

Le terme fractal a été introduit par Benoît MANDELBROT au cours des années 70, dans un ouvrage qui s'appelle Les Objets Fractals. Ce terme est formé à partir de l'adjectif latin *fractus* qui signifie irrégulier ou brisé. Benoît MANDELBROT est le "pilier" de la théorie des fractales qu'il a initié afin d'étudier les objets naturels qui ont en commun des formes extrêmement irrégulières ou interrompues [59]. Une fractale est un objet de complexité variable qui est décrit par la présence de détails à toutes les échelles, par une certaine auto-similarité (locale, globale ou statistique) et peut-être représenté par une définition relativement simple, parfois récursive. La géométrie fractale a pour objectif l'étude des fonctions et des phénomènes complexes qui ne peuvent pas être décrits par la géométrie euclidienne. Il faut noter que les objets étudiés par la géométrie fractale ne sont pas des objets d'exception, mais des objets naturels de la nature. La figure (Fig.2.2) illustre des exemples de fractales.

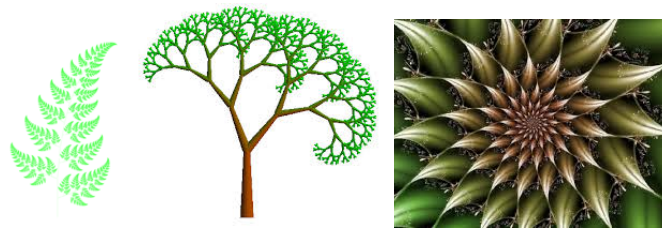


Figure 2.2 – Exemples de fractales

2.2.2 Principe des IFS

Un système de fonctions itérées (SFI), introduit pour la première fois en traitement d'images par M. Barnsley, permet de générer des images fractales [12, 13]. Il s'agit d'un ensemble de fonctions appelées transformations affines, qui ont la propriété d'être contractantes. Lorsque ces transformations sont appliquées avec plusieurs itérations, le SFI converge vers une image fractale fixe, nommée le point fixe. La figure (Fig.2.3) montre un exemple d'un SFI et de sa convergence vers le point fixe pour construire le triangle de Sierpinski. Évidemment, il y aura convergence seulement si un nombre suffisant d'itérations de la transformation globale est appliqué. Le fait de pouvoir générer une image



Figure 2.3 – Exemple d'un SFI

à partir d'un simple motif élémentaire suggère à l'application de cette méthode dans le

domaine de la compression d'image. En se basant sur la théorie des fractales, la compression d'une image consiste à générer un IFS approprié à cette image. La décompression se résume à l'application itérative de ce système sur une image jusqu'à l'obtention de l'image décodée après convergence.

Barnsley [13, 12] a démontré le théorème de collage qui a pour but de définir l'IFS pour une image donnée, ce théorème est la base de codage d'image par fractales. En 1989, Jacquin [45] a proposé une approche qui permet de coder une image naturelle par la méthode des fractales. Son principe est la génération d'une série de transformations affines contractantes et locales qui définit un opérateur global contractant. Dès 1989, de nombreux travaux ont vu le jour [46], [37], visant tous à améliorer l'approche de Jacquin pour optimiser les performances en matière de temps de calcul et du taux de compression. Donc, un IFS est un ensemble fini de transformations contractantes affines w_i . Le codage par fractales d'une image consiste à trouver un opérateur contractant W qui est l'union des transformations w_i .

$$W = \cup_{i=1}^m w_i \quad (2.4)$$

2.2.3 Approche de JACQUIN

Jacquin [44] a proposé un algorithme fractal pour coder les images naturelles. La figure (Fig.2.4) [16] décrit le principe de base de l'algorithme de Jacquin pour la compression fractale en utilisant des transformations affines locales w_i .

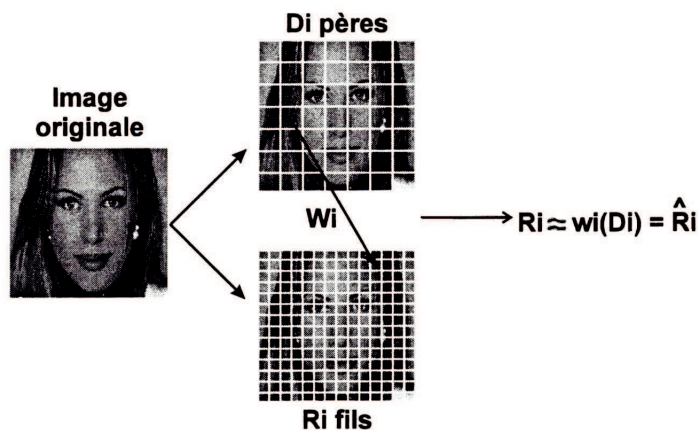


Figure 2.4 – Partitionnement en bloc source et bloc cibles [16]

L'image est partitionnée en blocs : des blocs sources d_i (blocs de tailles grandes) et des blocs cibles r_i (blocs de tailles petites). Les blocs cibles seront codés à partir des blocs sources, le partitionnement pour les blocs cibles doit recouvrir toute l'image sans qu'il ait de chevauchement entre eux. Par contre pour les blocs sources, le partitionnement est

quelconque et il n'y a pas de règle spécifique pour le faire. Pour chaque bloc cible r_i , il existe un bloc source d_i , de l'ensemble D des blocs sources, qui lui ressemble avec le moins d'erreurs possible. Bien entendu, une transformation affine locale w_i est appliquée sur un bloc source D_i pour le faire ressembler le plus possible au bloc cible r_i . L'union des transformations w_i forme la transformation globale W qui a pour unique point fixe, l'image initiale. Il s'agit donc de minimiser l'erreur entre l'image initiale et le point fixe. Il faut minimiser l'erreur pour chaque bloc cible. En général, la forme de la transformée affine locale w_i est la suivante :

$$r_i \simeq \hat{r}_i = w_i(d_i) = s_i(\sigma d_i) + o_i \quad (2.5)$$

Où σd_i est la contraction spatiale du bloc source pour avoir la même taille que le bloc cible. s_i est le facteur d'échelle photométrique (contraste) et o_i représente le facteur de décalage photométrique.

La contraction spatiale σ est une transformation linéaire qui agit sur le bloc source d_i de l'image pour l'amener au bloc cible r_i avec le moins d'erreurs possible. Cette transformation comporte les opérations suivantes : la réduction d'échelle, la rotation, la translation et la réflexion. Les paramètres x_i et y_i indiquent la position relative du bloc source par rapport au bloc cible (translation). L'opérateur de symétrie (sym_i) permet de renverser (effet miroir) et d'effectuer une rotation d'un multiple de 90 degrés sur le bloc source d_i . Le paramètre sym_i nécessite quelques bits pour l'encodage et permet d'améliorer la qualité de codage. Le quatrième paramètre est celui de la réduction d'échelle spatiale (décimation). La contraction photométrique (échelle ou décalage), doit préserver l'énergie alternative normalisée par la variance entre le bloc cible et le bloc source. L'ajustement de contraste s_i et de l'intensité lumineuse o_i peut modifier le bloc source pour minimiser l'erreur de codage. La figure (Fig.2.5) montre l'effet de ces deux facteurs sur le bloc source d_i pour l'amener au bloc cible. Par exemple, on remarque qu'un facteur de contraste s_i négatif appliqué avec un facteur de décalage o_i positif a un effet d'inversion sur les intensités de chaque quadrant.

La transformation globale W , formée par l'ensemble des transformations affines locales, est codée à l'aide de tous ces paramètres. Ce sont les paramètres qui seront quantifiés, compressés et transmis au décodeur.

Le calcul de l'erreur s'effectue à chaque comparaison avec la banque de recherche de blocs sources générée à l'avance. Pour effectuer la comparaison entre les deux types de blocs, on calcule une mesure de la distance quadratique moyenne :

$$d_{rms}(f, g) = \sqrt{\frac{1}{NM} \sum_{x=1}^N \sum_{y=1}^M (f(x, y) - g(x, y))^2} \quad (2.6)$$

Où f est le bloc cible et g est le bloc source contracté par l'opérateur σ . Les deux blocs ont deux tailles différentes N et M mais proviennent de la même image I^2 . La métrique

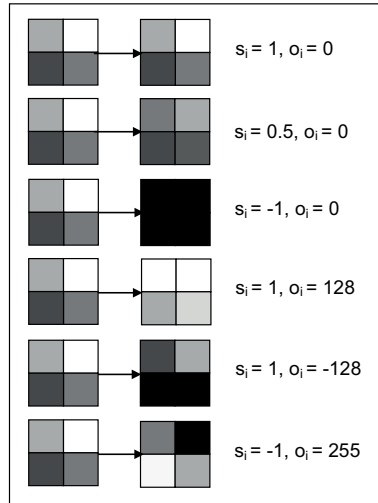


Figure 2.5 – Contraction photométrique sur le bloc source D_i

de la distance quadratique moyenne (éq.2.6) est une bonne approximation de l'erreur de codage et elle est facile à implémenter. Il existe d'autres métriques telle que la distance absolue moyenne (éq.2.7) et la valeur supérieure des distances entre chaque pixel (éq.2.8).

$$d_{rms}(f, g) = \frac{1}{NM} \sum_{x=1}^N \sum_{y=1}^M |f(x, y) - g(x, y)| \quad (2.7)$$

$$d_{sup}(f, g) = \sup_{(x,y) \in I^2} |f(x, y) - g(x, y)| \quad (2.8)$$

Dans la section suivante, en va parler des étapes additionnelles pour réaliser un codec d'images fixes.

2.2.4 Paramètres additionnels du codec d'image

2.2.4.1 Partitionnement de l'image

Il existe plusieurs méthodes de partitionnement, jusqu'à présent on a considéré que les blocs source et cible étaient de forme carrée et fixe. Mais il existe d'autres formes plus complexes et variables. Dans ce mémoire on présente une approche principale basée sur le partitionnement en quadtree. Il existe de nombreux travaux sur ce domaine et le lecteur peut voir les détails dans les références suivantes : [68],[37], [28].

Fisher dans son livre [37], explique les méthodes de partitionnement telles que le quadtree et le HV (Horizontal - Vertical). Il les a expérimenté sur une image et ont donné de très bon résultats en matière de débit et de distorsion.

Le partitionnement en quadtree a été proposé par Fisher [37] dans la compression fractale.

Ce partitionnement est un partitionnement rigide et il est intéressant puisqu'il offre une flexibilité au niveau des détails dans l'image en plus il n'est pas coûteux en temps de codage (Fig.2.6)[37]. Le partitionnement en quadtree est une méthode adaptative qui

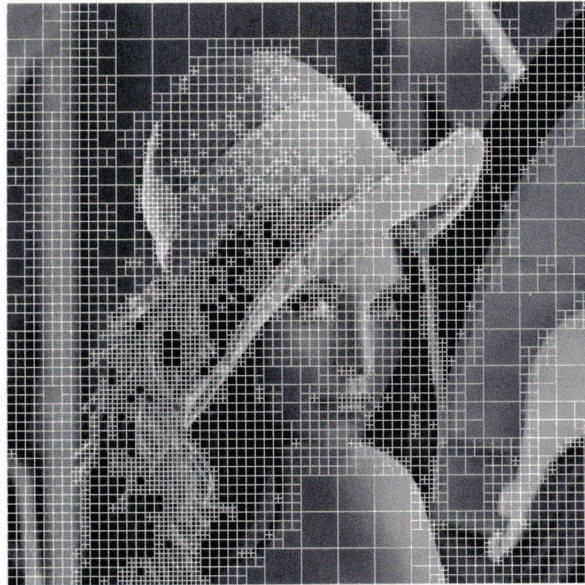


Figure 2.6 – Partitionnement en quadtrees

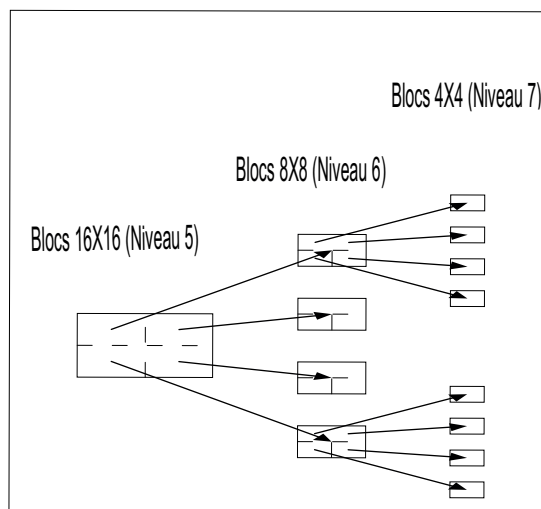


Figure 2.7 – Niveaux de partitionnement quadtree

dépend de l'erreur de codage d'un bloc. Pour les régions uniformes, il est plus facile de coder des gros blocs et de garder l'erreur minimale. Par contre, pour les régions plutôt détaillées, l'erreur de codage pour un gros bloc est trop élevée et on doit le partitionner en quatre sous-blocs. Le partitionnement continue jusqu'à ce que l'erreur de codage soit acceptable (dessous d'un seuil de tolérance) ou jusqu'à ce que le partitionnement maximal (la profondeur maximale du quadtree) soit atteint. La figure (Fig.2.7) montre le passage d'un niveau à l'autre dans le partitionnement d'une image 512X512.

Le tableau (Tab.2.1)[37] présente l'organigramme du partitionnement en quadtree traduit

du livre de Fisher selon un seuil de tolérance. Rappelons que cette méthode est celle implémentée dans notre codec.

-
- Choisir un seuil de tolérance e_c .
 - Pour l'ensemble des blocs cibles de l'image à la profondeur minimale.
 - {
 - Parmi les blocs sources D appartenant à la même classe que R_i , trouver D_i avec le w_i correspondant qui minimise l'erreur quadratique moyenne (éq. 2.6).
 - Si $d_{rms}(R_i, w_i(D_i)) < e_c$ ou la profondeur maximale est atteinte, alors
 - Envoyer la transformation w_i et passer au prochain R_i , remonter en profondeur s'il y a lieu.
 - Sinon partitionner R_i en quatre sous-blocs et coder chaque sous-bloc.
 - }
-

Table 2.1 – Organigramme du partitionnement en quadtree selon un seuil de tolérance

Au tableau (Tab.2.2)[37], on retrouve un autre code du partitionnement en quadtree selon un nombre total de transformations affines prédéterminé avant le partitionnement. Cette méthode vise à obtenir un taux de compression fixe.

-
- Choisir un nombre total de transformations N_r .
 - Pour l'ensemble des blocs cibles de l'image à la profondeur minimale
 - {
 - Parmi les blocs sources D appartenant à la même classe que r_i , trouver d_i avec le w_i correspondant qui minimise l'erreur quadratique moyenne (éq. 2.6).
 - ajouter dans une liste la transformation w_i , l'erreur $d_{rms}(r_i, w_i(d_i))$ et la position d_i .
 - }
 - Tant que le nombre de r_i dans la liste est plus petit que N_r
 - {
 - Parmi les r_i dans la liste qui ne se retrouvent pas à la profondeur maximale, trouver le r_j avec la mauvaise erreur.
 - Partitionner r_j en quatre sous-blocs, les coder et les ajouter à la liste.
 - Enlever r_j de la liste.
 - }
 - Envoyer les N_r transformations w_i dans la liste.
-

Table 2.2 – Organigramme du partitionnement en quadtree selon un nombre total de transformations.

2.2.4.2 Recherche de l'auto-similarité dans un voisinage limité

L'inspection des blocs sources dans la totalité de l'image est le principe de la compression par fractales. Si l'auto-similarité apparaît partout dans l'image alors cette méthode sera coûteuse en temps de calcul. En général, les images qu'on traite ne sont pas des images fractales mais présentent plus ou moins des ressemblances en particulier dans des voisinages limités. Donc pour un bloc cible r_i donné, on peut considérer une zone de $z \times z$ bloc source d_j pour la recherche de l'auto-similarité. Le tableau (Tab.2.3) montre les variations du $PSNR^2$ entre l'image et sa transformation, le taux de compression et le temps de calcul, en ne modifiant que la zone de recherche des blocs sources autour du bloc cible, pour Lena 512x512x8 bits. La taille des blocs cibles a été fixée à 4x4. On remarque que la limitation de la zone de recherche réduit le temps de calcul par contre le taux de compression augmente. Il faut donc trouver un compromis entre le débit et la distorsion.

Zone de $z \times z$ blocs voisins	PSNR(db)	Taux	Temps de calcul
8x8	32.25	5.55 (1.44)	19"
16x16	33.61	5.2 (1.53)	1'3"
32x32	34.83	4.83 (1.65)	4'18"
64x64	35.98	4.7 (1.77)	19'15"
128x128	37.13	4.2(1.9)	1 ^h 13'20"

Table 2.3 – Influence de la zone de recherche

2.2.4.3 Transformations contractantes affines locales LIFS

Dans cette section, on va revoir la théorie des IFS (Sec.2.2.2), en essayant de faire un résumé et de reformuler les équations pour qu'on puisse travailler avec dans le domaine transformé (Sec.3.2). Posons I l'intensité de l'image qu'on veut encoder, et $\mathbf{x} = (x, y)$ les coordonnées spatiales des pixels de cette image. Posons aussi, $\{r_1, \dots, r_N\}$ l'ensemble des blocs cibles (range block) qui ne chevauchent pas ; c'est-à-dire une collection de pixel qui partitionne l'image sans recouvrement entre les blocs. Semblablement, $\{d_1, \dots, d_M\}$ est un ensemble de M blocs source (Domain block) possiblement en chevauchement qui couvrent l'image. Pour permettre une notation compacte, I_{r_i} et I_{d_j} indiquent la restriction de l'intensité I au bloc cible r_i et au bloc source d_j respectivement. En d'autres termes, $I_{r_i} = \{I(\mathbf{x}) : \mathbf{x} \in r_i\}$ tandis que $I_{d_j} = \{I(\mathbf{x}) : \mathbf{x} \in d_j\}$.

Pour chaque bloc cible r_i l'objectif est de définir un bloc source d_j et un ensemble de transformations contractantes affines locales w_i qui minimise le critère de distorsion ε . La transformation contractante w_i comporte trois opérations : la contraction, la transformation photométrique et la transformation géométrique.

L'opération de contraction σ est appliquée sur les pixels du bloc source. Afin de réduire le chevauchement, un filtre passe bas, une simple moyenne, doit être effectuée avant la contraction. La moyenne des quatre voisins suivie par la contraction de K (sous-échantillonnage) est souvent utilisée :

$$I(\mathbf{x}) \stackrel{\sigma(I, \mathbf{x})}{\longleftarrow} \frac{1}{4} \sum_{\mathbf{y} \in \eta(K\mathbf{x})} I(\mathbf{y})$$

Où $\eta(\mathbf{x}) = \{(x, y), (x + 1, y), (x, y + 1), (x + 1, y + 1)\}$ est un voisinage non-symétrique de $\mathbf{x} \in r_i$. La substitution ci-dessus doit commencer à $\mathbf{x} = 0$ pour éviter la perte de données. La deuxième étape consiste à appliquer une transformation photométrique γ pour les différents pixels des blocs sources et cibles.

$$\gamma(I) \stackrel{\Delta}{=} sI + o$$

Où s est le facteur d'échelle (gain) et o est le facteur de décalage.

Dans la troisième étape, une transformation géométrique est effectuée par le moyen d'une transformation affine appliquée sur les blocs sources déjà contractés et transformés par la transformation photométrique (source \rightarrow cible). Cependant, lors du codage pour chaque $\mathbf{x} \in r_i$ on a besoin de trouver des paramètres de l'application réciproque (cible \rightarrow source). Cette transformation affine inverse peut être définie comme suit :

$$\tilde{\zeta}(\mathbf{x}) \stackrel{\Delta}{=} \mathbf{A}\mathbf{x} + \mathbf{b} \quad (2.9)$$

Où \mathbf{A} est une matrice 2x2 et \mathbf{b} est un vecteur de translation. Notons que cette transformation doit assurer une correspondance 1-à-1 entre les pixels des blocs cibles et des blocs sources. Par exemple, la matrice \mathbf{A} souvent comporte les opérations comme la rotation et la réflexion.

Avec les notations ci-dessus, la transformation w_i globale qui transforme les pixels du bloc source aux pixels du bloc cible en \mathbf{x} peut être exprimé comme suit :

$$w_i(I, \mathbf{x}) \stackrel{\Delta}{=} \gamma \circ \sigma(I, \tilde{\zeta}(\mathbf{x})), \quad \mathbf{x} \in r_i \quad (2.10)$$

Où \circ est l'opérateur de composition.

Comme mentionné ci-dessus, les paramètres de la transformation géométrique (éq.2.9) sont souvent limités à un petit ensemble. Dans le passé, les opérations isométriques, telles que la rotation de blocs et la réflexion, ont été souvent utilisées en matière d'une matrice \mathbf{A} . En outre, on limite les transformations \mathbf{A} aux huit cas suivants : quatre rotations ($0^\circ, 90^\circ, -90^\circ, 180^\circ$) et quatre réflexions (mi-horizontale, mi-vertical, première diagonale, deuxième diagonale) [45, 46]. On note l'ensemble des transformations possibles \mathbf{A} par $\{\zeta^p\}_{p=1}^8$. Cette transformation simplifiée peut-être écrite comme suit :

$$w_i(I, \mathbf{x}) \stackrel{\Delta}{=} s_i \cdot \sigma(I, \zeta_i^p(\mathbf{x}) + \mathbf{b}_i) + o_i, \quad \mathbf{x} \in r_i.$$

Puisque le vecteur de translation \mathbf{b}_i n'est pas arbitraire, mais limité aux positions indiquées par des blocs sources partiellement chevauchés, on peut exprimer \mathbf{b}_i implicitement

comme l'indice du bloc source, c'est-à-dire, j dans d_j . En utilisant cette notation modifiée, on peut redéfinir la transformation (éq.2.10) comme suit :

$$w_i(I_{d_j}, \mathbf{x}) \triangleq s_i \cdot \sigma(I_{d_j}, \zeta_i^p(\mathbf{x})) + o_i, \quad \mathbf{x} \in r_i. \quad (2.11)$$

Pour coder le bloc cible r_i , on doit faire une recherche de l'indice j (le bloc source d_j) et d'une isométrie ζ_i^p , en plus du calcul de paramètres photométriques s_i et o_i . Ceci peut être fait en minimisant l'erreur quadratique moyenne suivante :

$$\varepsilon(I_{r_i}, I_{d_j}, w_i) = \frac{1}{\text{Card}(r_i)} \sum_{\mathbf{x} \in r_i} [I_{r_i}(\mathbf{x}) - w_i(I_{d_j}, \mathbf{x})]^2. \quad (2.12)$$

Tandis que l'isométrie ζ_i^p et l'indice j (l'équivalent de la translation \mathbf{b}) sont estimés par une recherche approfondie, les facteurs d'échelle s_i et de décalage o_i sont calculés par les formules suivantes :

$$s_i = \frac{\sum_{\mathbf{x} \in r_i} [\sigma(I_{d_j}, \zeta_i^p(\mathbf{x})) - m_{d_i}] [I_{r_i}(\mathbf{x}) - m_{r_i}]}{\sum_{\mathbf{x} \in r_i} [\sigma(I_{d_j}, \zeta_i^p(\mathbf{x})) - m_{d_i}]^2}, \quad (2.13)$$

$$o_i = m_r - s_i \cdot m_d, \quad (2.14)$$

Où m_{r_i} et m_{d_i} sont les valeurs d'intensité moyennes pour les blocs cibles et les blocs sources, respectivement. Au lieu de transmettre le facteur de décalage photométrique o_i (en plus de j , ζ_i^p et s_i), la valeur moyenne m_{r_i} du bloc cible r_i peut être aussi transmise. Ceci permet la représentation précise de l'intensité moyenne locale et assure la convergence au décodeur (orthogonalisation) [66].

Un codage/décodage fractal typique est exécuté comme suit : pour chaque bloc cible r_i dans l'image, une recherche de la transformation optimale w_i est effectuée. Les paramètres w_i , qui minimisent la distorsion ε , sont stockés ensuite et codés par le codage entropique puis transmis. La procédure est itérée pour tous les blocs cibles. Les transformations de bloc locaux constituent l'ensemble de la transformation globale de l'image qui permet de trouver une approximation proche de cette image. Au décodeur, les paramètres de transformation sont reconstruits à partir des codes reçus et ensuite la transformation est itérativement appliquée à une image arbitraire. La convergence à l'image fractale désirée aboutit après un certain nombre d'itérations.

2.2.4.4 Classification des blocs

Plusieurs méthodes peuvent nous permettre de réduire le temps de codage dans le but d'utiliser les algorithmes en temps réel. Pour effectuer une telle tâche, on peut faire une classification des blocs sources, c'est-à-dire regrouper les blocs qui se ressemblent dans un même sous-groupe représenté par une forme canonique et une plage de variance. La figure (Fig.2.8)[37] montre les trois formes canoniques de base d'un bloc divisé en quatre quadrants représenté par la moyenne de chaque pixel à l'intérieur du quadrant



Figure 2.8 – Les trois formes canoniques selon la moyenne.

correspondant.

Pour que chaque bloc puisse être classifié parmi une de ces trois formes, on doit parfois appliquer un renversement (effet miroir) des quadrants (Fig.2.9).

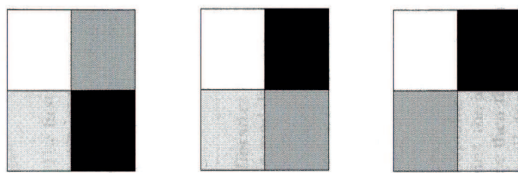


Figure 2.9 – Les trois formes canoniques renversées

La moyenne de chaque quadrant i se calcul de la façon suivante :

$$A_i = \frac{1}{n} \sum_{j=1}^n r_j^i \quad (2.15)$$

où r_j^i est le j^{ime} pixel du cadrant i , et n est le nombre de pixels pour chaque quadrant. Ensuite, une rotation est effectuée pour que l'intensité moyenne la plus élevée soit associée à A_1 et les autres A_i représentent l'une des trois formes canoniques suivantes :

$$A_1 \geq A_2 \geq A_3 \geq A_4$$

$$A_1 \geq A_2 \geq A_4 \geq A_3$$

$$A_1 \geq A_4 \geq A_2 \geq A_3$$

Pour chacune des 3 formes canoniques, il y a 24 possibilités de classification de la variance pour chaque quadrant i . Le calcul est le suivant :

$$V_i = \frac{1}{n} \sum_{j=1}^n (r_j^i - A_i)^2 = \frac{1}{n} \sum_{j=1}^n (r_j^i)^2 - A_i^2 \quad (2.16)$$

En classifiant les blocs, le temps de recherche pour chaque bloc cible est réduit puisque le nombre de membres dans chaque classe devrait être considérablement inférieur au nombre total de blocs sources dans l'image. L'ensemble des blocs sources constitue la zone de recherche globale d'une image. La classification permet donc de diviser cette zone de recherche selon les différentes textures retrouvées dans l'image.

2.2.5 Décodage de JACQUIN

Au décodeur, il s'agit d'appliquer les transformations w_i reçues à une image arbitraire avec plusieurs itérations pour éventuellement converger vers un point fixe qui représente l'image codée avec un degré de distorsion. Pour que le décodage converge, les transformations affines w_i doivent être contractantes. Pour s'assurer que l'ensemble des transformations soit contractant, il faut rendre chaque transformation w_i contractante. Jacquin a démontré que pour qu'une transformation affine soit contractante, la valeur absolue de s_i , le facteur d'échelle de la transformation photométrique, doit être inférieur à 1. De plus, on suggère que chaque bloc cible ait une taille inférieure ou égale à celle du bloc source qui lui correspond. La figure (Fig.2.10)[45] illustre un exemple de décodage itératif de Lena. Le processus de décompression est initialisé avec une image uniforme. L'image



Figure 2.10 – *Décodage de l'image de Lena [16]*

reconstruite est obtenue dans sa qualité finale après 8 itérations et on peut constater que dès la première itération, il est possible de reconnaître l'image. Normalement, en partant d'une image uniforme, la première itération nous donne l'allure du partitionnement avec des blocs de tailles différentes (pour le quadtree) et d'intensité uniforme (la moyenne).

La présente section nous a introduits au codage fractal d'image. On a étudié la théorie des fractales de même que son utilité pour coder les images fixes. Dans la section suivante, on devra considérer la troisième dimension retrouvée dans les séquences vidéo, la composante temporelle. On détaillera les algorithmes nécessaires existant pour effectuer le codage de la vidéo.

2.3 CODAGE DES SÉQUENCES VIDÉO

La vidéo contient en plus des deux composantes spatiales de l'image, une composante dynamique et temporelle. Le codage de la vidéo est basé sur le codage de l'image, donc il faut bien comprendre les propriétés de la vidéo avant d'adapter un codec d'images fixes au codage de séquences.

La vidéo est tout simplement une séquence d'images, nommées trames (frames en anglais), espacées d'une période de temps, entre $\frac{1}{30}$ et $\frac{1}{25}$ secondes pour la télévision et le multimédia. Une fréquence de 25Hz (soit 25 trames par seconde), suffit pour donner l'impression au système visuel humain qu'il s'agit d'une séquence continue d'image comme s'il le percevait dans la réalité.

Notre contribution à la recherche consiste à implémenter un algorithme de compression de séquences vidéo par région à l'aide de la méthode des fractales dans le domaine fréquentiel. Mais avant de détailler notre approche (Chapitre 3), il est important de bien comprendre la méthode de [18], qui représente un algorithme de compression de séquences vidéo par la méthode des fractales dans le domaine spatial. Cet algorithme utilise comme outil de départ le code source [37] qui applique le codage fractal à une image fixe puis il a été modifié pour manipuler plusieurs images à la fois en exploitant les propriétés de la dimension temporelle. Le code source de [37] utilise un partitionnement de l'image en quadtree. Comme on a vu précédemment ; le partitionnement en quadtree est intéressant car il est facile à implémenter, son faible coût en temps de calcul et sa flexibilité. C'est un partitionnement adaptatif, il permet d'utiliser des petits blocs pour coder les régions détaillées et dynamiques. Inversement, il permet d'utiliser des grands blocs pour coder les régions uniformes et stationnaires.

2.3.1 Codage Vidéo

Une séquence vidéo est composée de trois dimensions. Les deux premières sont spatiales et forment l'image fixe, alors que la troisième est temporelle et sépare les trames dans la séquence. La figure (Fig.2.11) illustre les composantes d'une séquence vidéo.

Dans le codage de la vidéo on utilise généralement le concept du codage prédictif. Dans

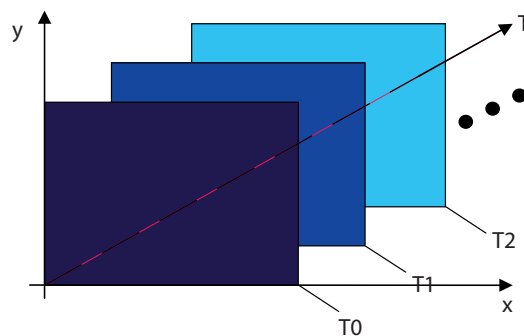


Figure 2.11 – Composantes d'une séquence vidéo

une séquence continue, si la période entre deux trames est faible, alors les trames consécutives sont similaires à l'exception du mouvement de certains objets. Ce mouvement est donc facile à prédire. En fait, plus la période entre deux trames est faible, plus la corrélation temporelle est élevée et plus le codage sera fidèle.

Il existe plusieurs techniques d'estimation de mouvement qui consistent à étudier le déplacement des objets dans une séquence vidéo, en cherchant la corrélation entre deux images successives afin de prédire le changement de position du contenu. L'estimation de mouvement est suivie par la compensation de mouvement qui a pour but d'appliquer les vecteurs de mouvement trouvés à l'image afin de synthétiser la transformation de la prochaine image. Tous les algorithmes utilisés sont basés sur l'exploitation de la corrélation temporelle entre les trames [91].

2.3.2 Estimation de mouvement par la méthode des fractales

Le concept d'estimation de mouvement par l'approche des fractales pour un bloc cible consiste à trouver un bloc source qui lui ressemble dans la trame précédente après avoir subi une transformation [20]. Les facteurs d'échelle et de décalage s et o sont optimisés à chaque comparaison pour minimiser l'erreur quadratique moyenne (éq.2.6). Un vecteur de mouvement $a(x,y)$ indique la position relative du meilleur bloc source. Au lieu de coder l'erreur et la transmettre, on la considère en effectuant le décodage de la trame actuelle. La trame suivante sera codée à partir d'une trame décodée qui inclut l'erreur. La figure (Fig.2.12) illustre le processus de l'estimation de mouvement par la méthode des fractales.

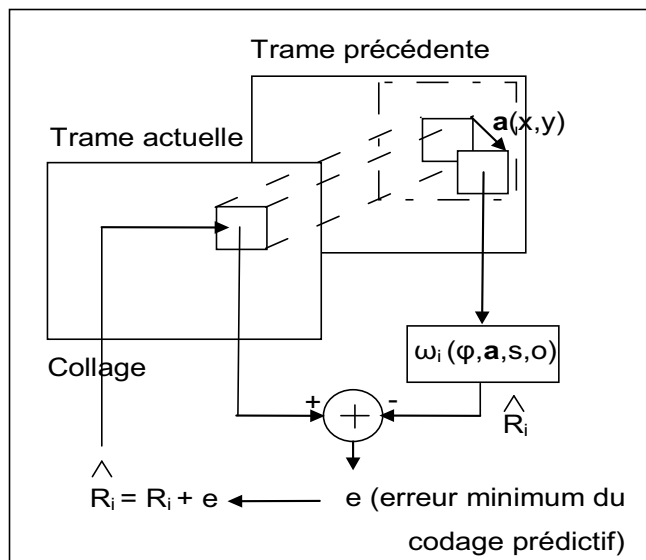


Figure 2.12 – Estimation de mouvement par la méthode des fractales

2.3.3 Approches pour le codage vidéo

Il existe dans la littérature deux algorithmes nécessaires pour effectuer la compression fractale d'une séquence vidéo. Le premier algorithme est le codage circulaire connu sous le nom "Circular Prediction Mapping" (CPM) et le deuxième algorithme est le codage prédictif connu sous le terme "Noncontractive Interframe Mapping" (NCIM). C.S Kim et S.U Lee [51] et [52] sont ceux crédités d'avoir développé ces méthodes.

2.3.3.1 Codage contractif et circulaire de prédiction (CPM)

Un nombre prédéterminé de (M) trames est chargé en mémoire. Le codage fractal a lieu entre deux trames consécutives en s'orientant toujours dans le même sens. Dans la figure (Fig.2.13) les M premières trames sont considérées comme un groupe initial et chaque trame sont prédite, d'une manière circulaire, à partir de celle qui la précède. La

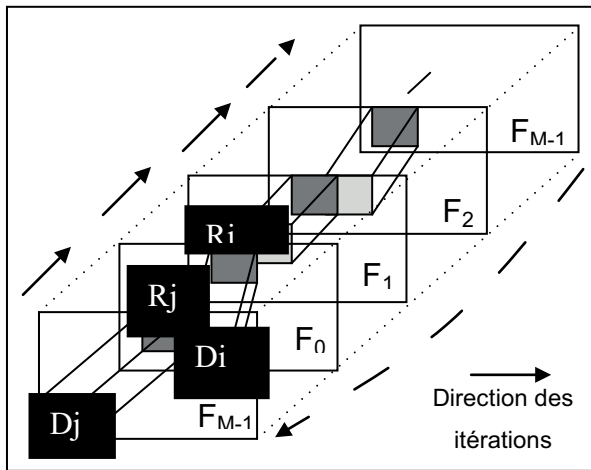


Figure 2.13 – Codage CPM pour M trames consécutives

trame F_1 est codée à partir de la trame F_0 . La trame F_2 est ensuite codée à partir de la trame F_1 . Pour fermer le cercle, la trame F_0 est codée à partir de la trame F_{M-1} .

Les blocs cibles sont codés à partir des blocs sources de même taille que dans la trame précédente. Ainsi, les transformations affines ne contiennent plus d'opérateur de contraction spatiale. L'équation (éq.2.5) devient alors de la forme suivante :

$$R_i \approx \hat{R}_i = s_i(D_{a_i}) + o_i \cdot C \quad (2.17)$$

Où a_i est la position relative du bloc source D_i dans la trame précédente par rapport à R_i , s_i est le facteur d'échelle photométrique (contraste), o_i est le facteur de décalage photométrique (intensité) et C est un bloc uniforme de valeurs unitaires. Le facteur d'échelle photométrique (contraste) doit toujours respecter la limite $|s_i| < 1$.

Dans la section (Sec.2.2.4.2), on a vu que la recherche du bloc source dans un voisinage limité permet de réduire le temps de codage. Dans le codage de la vidéo on distingue

deux méthodes de recherche dans un voisinage limité (recherche dite locale), la recherche en spirale et la recherche en mode de balayage X-Y. Dans la recherche en spirale, le bloc source est initialement collé au bloc cible, c'est-à-dire le vecteur de mouvement $a(x, y)$ est de $(0, 0)$. Ensuite, l'erreur est minimisée en appliquant une faible translation sur le vecteur de mouvement $a(x, y)$ pour rechercher le bloc source le plus semblable au bloc cible. Pour le mode de balayage X-Y la région de recherche est balayée à partir du coin haut de gauche à droite et de haut en bas.

L'approche [18] exploite la méthode de balayage X-Y, cette méthode est facile à implémenter mais elle est coûteuse en temps de codage. Par contre la recherche en spirale minimise considérablement le temps de codage puisque la probabilité d'obtenir $a(0, 0)$ est la plus élevée et elle constitue le premier choix de recherche. La figure (Fig.2.14) illustre ces deux méthodes.

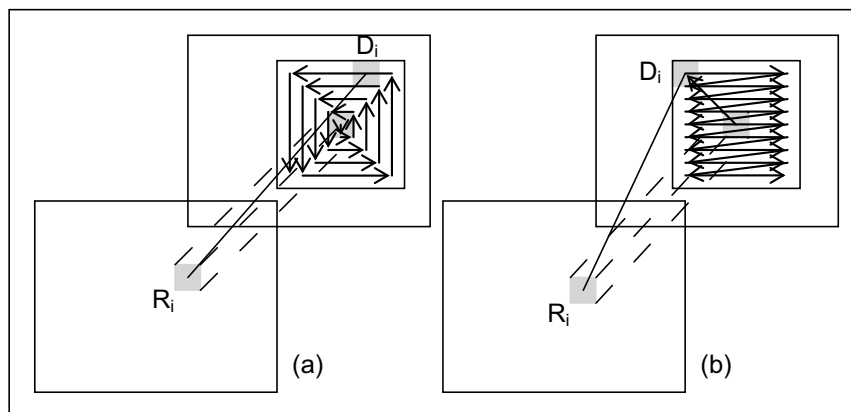


Figure 2.14 – Recherche locale : (a) Spirale, (b) Balayage X-Y

Le codage fractal inter-trame s'effectue de la même façon que celui d'une image fixe. Les calculs de l'erreur et des paramètres s_i et o_i s'effectuent de façons semblables. Au décodage, les transformations seront appliquées à un ensemble de M trames arbitraires, une trame à la suite de l'autre. La figure (Fig.2.15) montre l'application de N itérations circulaires pour effectuer le décodage CPM.

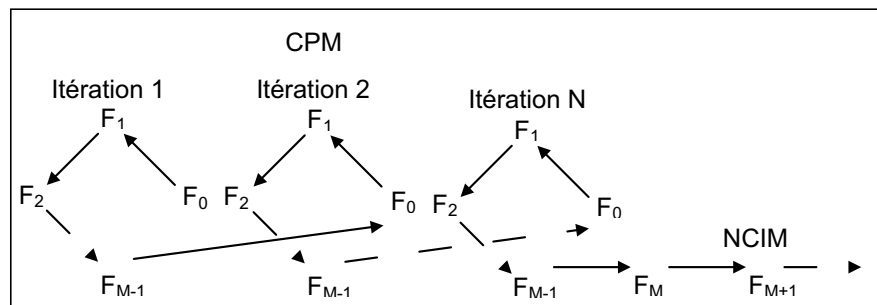


Figure 2.15 – Décodage CPM

2.3.3.2 Codage non-contractif de prédiction (NCIM)

Le codage NCIM effectue l'estimation de mouvement entre deux trames consécutives. Ce codage nécessite une trame entièrement décodée pour qu'il réussisse à coder/décoder la trame suivante. Le décodage n'est plus itératif comme il l'était dans le CPM. Les transformations du NCIM sont semblables à celles du CPM à l'exception qu'elles ne sont pas essentiellement contractantes. Le terme "non contractive" signifie que le facteur d'échelle photométrique s_i ne vérifie pas $|s_i| < 1$ comme dans le cas du CPM. Bref, le CPM doit être contractant pour assurer la convergence du processus itératif.

Une autre différence entre le CPM et le NCIM est au niveau du décodage. Le CPM doit effectuer plusieurs itérations pour converger vers un point fixe pour reconstruire les M premières trames. Alors que le NCIM décode les trames restantes avec une seule itération. Ceci permet de réduire le temps de décodage.

2.3.3.3 Codec hybride CPM/NCIM

C.S Kim et S.U Lee [51] et [52] ont proposé un algorithme hybride composé des deux algorithmes de base, le CPM et le NCIM, voir (Fig.2.16)[51].

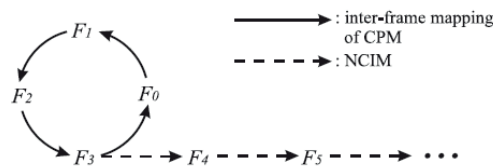


Figure 2.16 – Structure hybride du CPM et NCIM [51]

Dans la figure (Fig.2.16), les quatre premières trames F_k ($0 \leq k < 4$) sont codées par l'algorithme CPM et le reste des trames F_k ($k \geq 4$) sont codées par l'algorithme NCIM. Après le décodage du CPM, le NCIM commence à partir de la dernière trame décodée pour coder les trames qui succèdent. Il est important d'effectuer le décodage de chaque trame du NCIM au fur et à mesure dans l'encodeur pour éviter l'accumulation d'erreur dans les séquences continues. Les deux figures suivantes (Fig.2.17a)(Fig.2.17b) représentent les organigrammes du codeur et du décodeur hybride CPM/NCIM.

La variable T représente le numéro de trame à partir du début d'un codage/décodage CPM. Les trames de 0 à $K - 1$ sont celles de l'ensemble CPM et les trames avec $T \geq K$ sont celles de l'ensemble NCIM. Lorsqu'un changement de scène est détecté, la trame de la nouvelle scène prend la valeur de $T = 0$ et le codage/décodage CPM recommence.

Les trames de l'ensemble du CPM doivent avoir une corrélation temporelle positive entre elles. La raison pour laquelle le CPM ne peut pas coder les changements brusques de scène. Un algorithme supplémentaire doit être implémenté pour détecter le changement

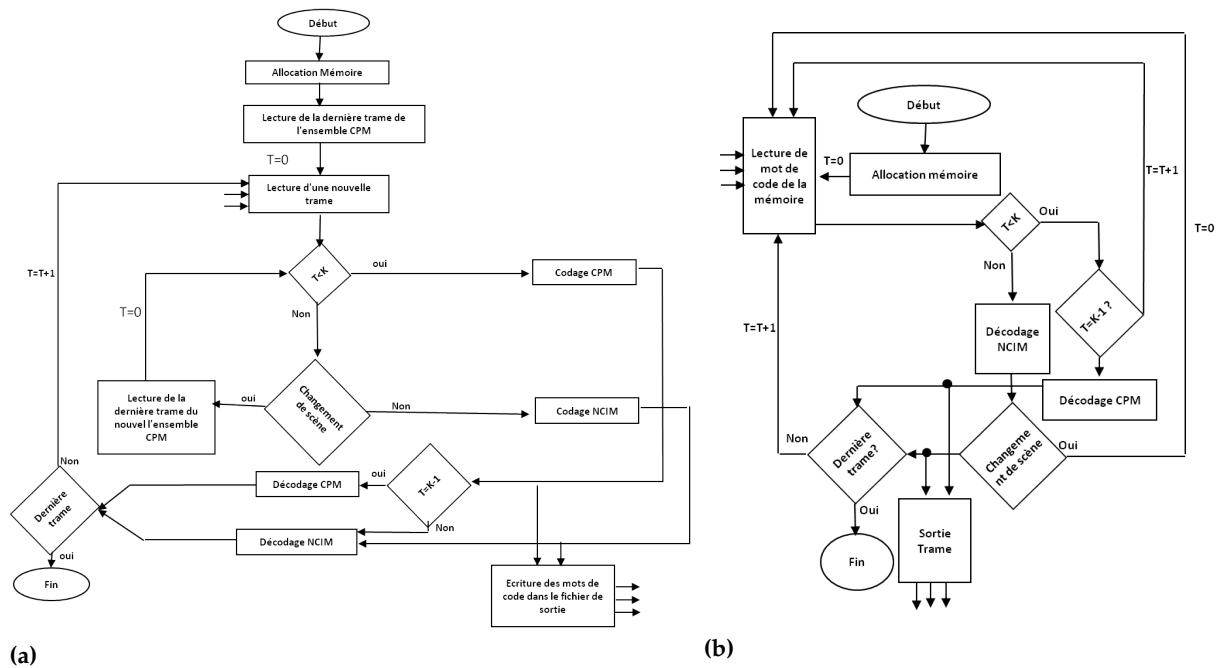


Figure 2.17 – Codec hybride CPM/NCIM : (a) Codeur CPM/NCIM, (b) Décodeur CPM/NCIM

de scène et ceci permet de se diriger vers le codage nécessaire voir (Fig.2.18).

2.3.4 Structure du codec fractal des séquences vidéo dans le domaine spatial

L'objectif est de réaliser un codec vidéo basé-région dans le domaine fréquentiel, mais avant de détailler notre méthode (Chapitre.3), il est important de bien comprendre l'approche de base [18].

L'approche de base est un programme écrit en langage C, l'auteur a converti le code C du codeur fractal d'une image fixe [37], en codeur de séquence d'images. Ce codec lit un ensemble de K trames consécutives en mémoire et effectue le codage circulaire (CPM). L'auteur a modifié le code de l'image fixe au niveau du codage des blocs cibles pour que les blocs sources associés soient de la même taille. La recherche des blocs sources se fait en mode de balayage X-Y et le partitionnement est un partitionnement en quadtree. Chaque trame des K trames de l'ensemble CPM doit avoir une corrélation temporelle positive avec celle qui la précède, car le CPM ne peut pas coder les changements brusques de scène. La vérification du changement de scène se fait par l'implémentation d'un bloc supplémentaire qui doit assurer la vérification de la corrélation entre deux trames consécutives (Fig.2.18). Après le décodage du CPM, le NCIM commence à partir de la dernière trame décodée pour coder les trames restantes.

2.3.4.1 Structure des séquences vidéo

Le codage de cette approche est effectué sur la composante Y, comme c'est le cas pour la plupart des méthodes existantes. La composante Y est codée en format niveaux de gris

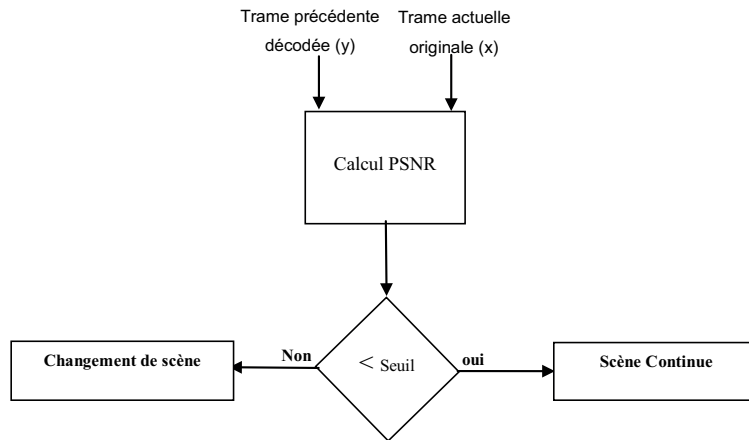


Figure 2.18 – *Changement de scène*

sur 8 bits. Les séquences vidéo utilisées sont sous le format Vids ou Telenor.

Les séquences de format Vids comportent deux fichiers. Le premier fichier est du type descripteur (*.des) qui contient des informations sur la séquence. Dans ce fichier on trouve le nombre de trames, les dimensions de la vidéo, la fréquence d'échantillonnage, et d'autres informations qui servent à décrire la séquence d'images. Tandis que le deuxième fichier est de type données (*.pic) qui contient les valeurs de pixels pour chaque composante. Le programme en C de cette approche prend en charge la lecture des séquences en format vids. Il permet aussi l'écriture des séquences décodées en format vids en exploitant les bibliothèques existantes qui permettent l'ouverture, la lecture, l'écriture et la fermeture de ce type de format.

2.3.4.2 Allocation mémoire

Cette approche permet la gestion de la mémoire par l'utilisation des structures de données de types listes chaînées. Ceci réduit l'utilisation de la mémoire par le processus d'allocation et de libération que procurent les listes chaînées selon le besoin dynamique des algorithmes. En fait, seulement les K trames du codage CPM seront chargées en mémoire. Pour chaque trame chargée en mémoire, une structure sera allouée pour stocker les transformations affines locales. Donc, les K structures servent pour le codage CPM comme pour celui du NCIM.

2.3.4.3 Paramètres du codage

Dans cet algorithme on trouve deux types de paramètres avec lesquels on peut varier, ceux de l'algorithme fractal et ceux de l'implémentation du codec hybride.

1. Paramètres de l'algorithme fractal

Il existe plusieurs paramètres qui peuvent être ajustés pour trouver un compromis entre le taux de compression, le rapport signal-bruit et le temps d'encodage. Dans ce qui suit, on va citer les paramètres fractals qui affectent le fonctionnement du codec.

- Nombre de bits utilisés pour quantifier les facteurs d'échelle s_i .
- Nombre de bits utilisés pour quantifier les facteurs de décalage o_i .
- Profondeur minimum du partitionnement en quadtree.
- Profondeur maximum du partitionnement en quadtree.
- Utilisation de la classification des blocs sources selon la disposition de la moyenne dans chaque quadrant (3 classes) (Fig.2.8),(Fig.2.9).
- Utilisation de la classification des blocs sources selon la disposition de la variance dans chaque quadrant (24 classes) (éq.2.16).
- Valeurs absolues du facteur d'échelle s_i . Cette valeur doit être contractante pour le CPM, $|s_i| < 1$, et non contractante pour le NCIM.
- Tolérance d'encodage détermine la qualité de l'encodage et par conséquent, influence le taux de compression.

2. Paramètres d'implémentation du codec hybride

Il existe plusieurs paramètres qui ont servi pour l'implémentation du codec hybride CPM/NCIM [18]. Ces paramètres sont indépendants des paramètres de l'algorithme fractal et peuvent affecter significativement la fonctionnalité et la performance du codec. Parmi ces paramètres on trouve :

- Le paramètre K qui représente le nombre de trames dans l'ensemble CPM. Il peut être ajusté pour améliorer les résultats du codec CPM.
- La taille de la fenêtre de recherche locale. Ce paramètre est très intéressant car il permet de réduire considérablement le temps de codage.
- Facteur de sous échantillonnage temporel des trames.
- Un paramètre qui permet d'ajuster le programme pour faire la recherche globale ou bien locale.
- Un paramètre de codage entropique sur les transformations de chaque trame (Huffman).
- Un paramètre qui permet l'application ou non du codage différentiel pour le facteur de décalage de l'intensité o_i .

Tous ces paramètres sont ajustés et optimisés par la réalisation des tests pour y ressortir une structure d'implémentation optimale. Les résultats trouvés par ce codec sont comparables à ceux retrouvés dans les articles du même sujet [51], [52].

2.3.5 Structure du codec fractal basé-région des séquences vidéo dans le domaine spatial

Jusqu'à présent, on a parlé d'un codage qui s'effectue sur une séquence entière sans notion d'objet (Fixe ou en mouvement) et arrière-plan. Une nouvelle fonctionnalité a été

introduite au codec fractal hybride CPM/NCIM dans le domaine spatial [18]. Elle est basée sur une segmentation en région et elle permet de coder les objets dans les images de la séquence de façon indépendante. Cette méthode offre la possibilité de contrôler la largeur de bande de transmission et permet de mieux coder les frontières de différentes régions. [19] et [17] ont proposé une méthode de codage basée-région sur une image fixe dans le domaine fréquentiel en utilisant la DCT de forme adaptative SA-DCT (Shape Adaptive DCT) (Fig.3.1), et dans le domaine spatial, sans la transformation DCT, en utilisant l'algorithme SA-spatial.

Dans [18], l'algorithme du SA-spatial a été adapté au codec hybride CPM/NCIM pour séparer les objets et pour les coder indépendamment. La méthode basée-région a amélioré significativement la qualité du codec, mais le temps de codage a sensiblement augmenté car on a plus de bloc cible à coder (les blocs cibles sur les frontières des objets ont quasiment doublé).

Le bénéfice de la méthode basé-région pour une séquence d'image est l'exploitation de la corrélation élevée entre deux trames consécutives. Par exemple, si un objet reste fixe pendant plusieurs trames, il s'agit de le coder une seule fois depuis son apparition, et de coder, de façon indépendante, les objets en mouvement. La figure (Fig.2.19) contient deux trames consécutives de la séquence *Children* qui montre les enfants, le ballon et les lettres en mouvement. Par contre, l'arrière-plan est totalement figé et sans mouvement, donc très corrélé entre chaque trame. Alors, l'arrière-plan est codé une seule fois et les objets en mouvement sont codés indépendamment.

La méthode basée-région nécessite une segmentation priori des séquences d'images en



Figure 2.19 – Deux trames consécutives de séquence *Children* 10HZ

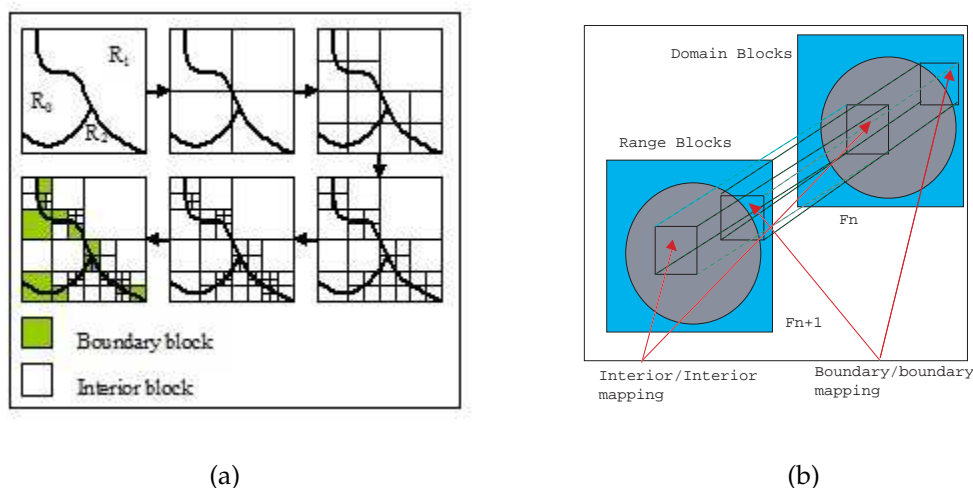
région, les objets en mouvement et l'arrière-plan. Le codage s'effectue sur les objets en mouvement. Les autres régions qui sont fixes ne sont codées qu'une seule fois dès le début de leur existence.

Pour délimiter les régions et leurs frontières, on utilise les alpha-planes comme c'est le cas dans le MPEG-4 [61]. La figure (Fig.2.20) montre un exemple de deux trames consécutives d'un alpha-plane pour la séquence *Children* vue dans la figure (Fig.2.19). Les alpha-planes binaires, représentant deux régions seulement, sont compressés par un fort taux de compression qu'il est négligé sur le calcul de la largeur de bande nécessaire. Par contre, les alphas-planes à plus de deux régions sont plus complexes et doivent être représentés par des valeurs entre 0 et 255.



Figure 2.20 – Deux trames consécutives d'un alpha-plane Children

Le codage de la région peut se faire de la même façon que dans l'encodeur ordinaire. Le partitionnement en quadtree reste toujours valable dans le basé-région. Il doit avoir un partitionnement adéquat surtout près des frontières de la région en déformation. Les blocs des frontières sont codés avec l'objet et avec l'arrière-plan pour représenter chaque région indépendamment. La figure (Fig.2.21(a)) illustre un partitionnement en quadtree près des frontières entre deux régions.



(a) (b)
Figure 2.21 – Illustration schématique du modèle du codage fractal de la vidéo basé-région. (a) Illustration du partitionnement en quadtree pour les blocs intérieurs et les blocs frontières, le même partitionnement est utilisé pour les blocs sources et cibles. (b) Des blocs cibles Intérieurs sont associés aux blocs sources intérieurs de la même région tandis que les blocs frontières sont partiellement correspondus aux blocs sources de la même-région.

On distingue deux types de codage, le codage des blocs frontières et le codage des blocs pleins. Le codage d'un bloc frontière cible se fait à partir d'un autre bloc frontière source qui contient un segment de la même région. De la même façon, le codage d'un bloc plein cible se fait à partir d'un autre bloc plein source de la même région. De cette façon, l'encodage et le décodage restent indépendants pour chaque région. Le domaine de recherche d'un bloc frontière est l'ensemble des blocs frontière dans tout l'objet concerné et le domaine de recherche d'un bloc intérieur est l'ensemble des blocs intérieurs dans tout l'objet concerné. La figure (Fig.2.21(b)) montre un exemple de recherche des blocs

pleins et des blocs frontières. Le codage des blocs de frontières se fait par l'algorithme du SA-spatial. L'application du SA-spatial nécessite la détermination des pixels appartenant à la région à encoder dans les blocs de frontières. Pour se faire, il s'agit de faire une recherche dans le alpha-plane et de faire ressortir les pixels dont le niveau de gris correspond à la région à encoder. On peut conclure que le alpha-plane est indispensable pour effectuer le codage basé-région.

La figure(Fig.2.22) illustre un exemple d'application de l'algorithme du SA-spatial. Les

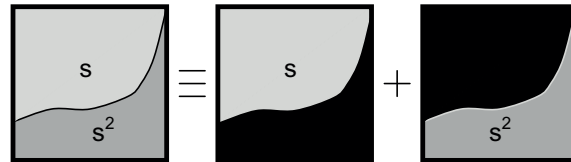


Figure 2.22 – Exemple d'application du SA-spatial

pixels appartenant à la région voisine seront temporairement substitués par la moyenne des pixels appartenant à la région considérée. On connaît cette procédure par le terme d'extrapolation. Pour adoucir la transition entre les pixels de la région et les pixels extrapolés, un filtrage d'ordre 2 est appliqué au bloc pour ne pas trop affecter les détails retrouvés dans les pixels actifs, mais il crée un effet d'étalement dans la zone où l'extrapolation a eu lieu. Le filtrage est réalisé à l'aide d'un filtre linéaire de phase et non causal. Il est effectué horizontalement à l'intérieur de chaque ligne puis verticalement à l'intérieur de chaque colonne.

Le segment fait partie maintenant d'un bloc entier. Ensuite, il s'agit d'effectuer une comparaison pixel par pixel en se servant du alpha-plane actuel (cible) comme masque. La figure (Fig.2.23) illustre un exemple de codage d'un segment cible à partir d'un segment de la même région (extrapolé et filtré) dans un bloc source frontière.

Le codage basé-région permet de coder l'arrière-plan une seule fois dès son apparition et tant qu'il reste fixe. Ceci permet de réduire la largeur de bande et le temps de codage. Par contre, les objets en mouvement affectent l'arrière-plan en laissant un changement derrière où ils se trouvaient dans la trame précédente. Des techniques de régénération doivent être implémentés pour mettre à jour ces changements qui ont eu lieu dans l'arrière-plan. Cette partie qui a subi un changement, doit être codée séparément et transmise pour effectuer la mise à jour au décodeur. La différence entre les deux trames successives de l'alpha-plane permet de déterminer les endroits de l'arrière-plan, qui ont subi un changement, nécessitant une mise à jour.

Dans cette partie, on a survolé les principaux points de la fonctionnalité de codage basée-région dans le domaine spatial. On trouve plus de détails sur cette fonctionnalité au chapitre (Chapitre 3). Les résultats obtenus dans [18] démontrent que des débits très faibles peuvent être atteints par l'application de cette fonctionnalité et une forte amélioration au niveau de la qualité surtout près des frontières comparativement au codec non basé-région. Puisque le codage de chaque objet est indépendant, il est maintenant possible

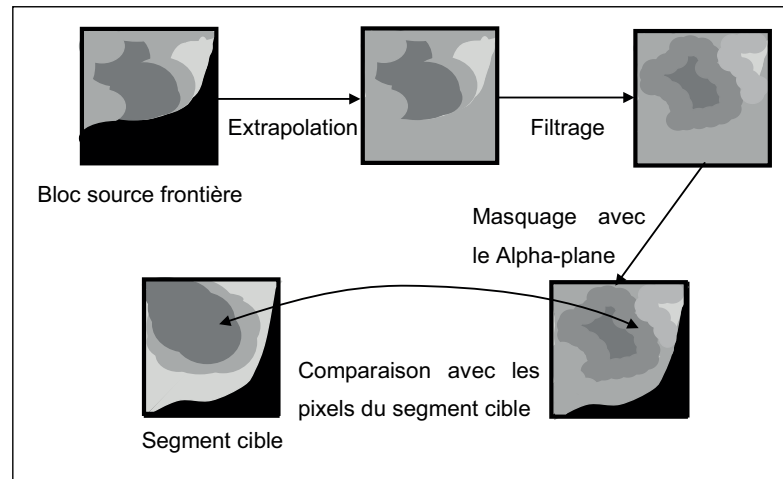


Figure 2.23 – *Processus de codage d'un bloc cible (Extrapolation et filtrage)*

de diminuer le taux de transmission des objets qui subissent très peu de mouvements ou de déformation.

2.4 CONCLUSION

Le présent chapitre nous a introduits au codage de l'image fixe. On a étudié la théorie des fractales de même que leurs utilités pour coder les images. Entre autres, la classification et le partitionnement adaptatif sont des fonctions qui permettent d'économiser la largeur de bande ainsi que le temps de codage sans trop affecter la qualité. Ensuite, on a introduit les algorithmes du codage de la vidéo par fractals. On a expliqué en détail, les algorithmes du CPM et du NCIM. On a détaillé l'algorithme de base de notre approche qui est présentée sous forme d'un algorithme hybride CPM/NCIM. Enfin, on a parlé brièvement de l'introduction de la fonctionnalité basée-région, son principe et ces avantages dans le codage de la vidéo. Dans le chapitre suivant, on détaillera le codage de la vidéo basé-région dans le domaine fréquentiel. On présentera les résultats trouvés et on les analysera.

CODAGE FRACTAL BASÉ-RÉGION DES SÉQUENCES VIDÉO UTILISANT LA SA-DCT

3

Basé sur une méthode classique de compression fractale d'image, une nouvelle méthode basée-région de compression vidéo utilisant le shape-adaptive DCT a été proposée dans ce chapitre. Premièrement, on utilise le programme de partitionnement en quadtree ; et une méthode de recherche locale, ceci réduit la stratégie de recherche du bloc. Le codage fractal est adopté et chaque objet est encodé indépendamment par une segmentation préalable de la vidéo en alpha-plane, comme il a été défini exactement en MPEG-4. Les résultats trouvés sont prometteurs et offrent un faible débit binaire.

3.1 INTRODUCTION

Deux approches majeures sont utilisées dans le codage de la vidéo : les approches basées-blocs et celles basées-régions. L'approche basée-blocs a l'avantage d'être simple, efficace et robuste. Elle a obtenu un grand succès dans les années précédentes et elle a été adoptée dans de nombreuses normes de compression vidéo (MPEG 1 / 2, H263, H264). Toutefois, l'inconvénient majeur de cette approche est la qualité subjective des images reconstruites à faible débit binaire. Le codage de séquence vidéo basé-région a été intensément étudié dans les dix dernières années et est également soutenu par la norme MPEG-4 [71, 61]. Il présente un avantage important par rapport au codage basé-blocs : il permet la manipulation d'objets sans décodage complet de l'image, ceci améliore la qualité de codage et réduit le débit binaire. Dans un tel schéma, une segmentation a priori (alpha-plane) [67] de l'image en objets, doit être connue à l'avance [26]. L'approche basée-régions a été considérée comme une alternative très prometteuse par rapport à l'approche basée-blocs. Elle diminue le problème des effets gênants de codage, tel que l'effet de blocs, en particulier lorsque les blocs coïncident avec les frontières de différents objets. Elle fournit également des représentations plus naturelles de scènes. Son avantage majeur est l'acquisition de l'information profonde, significative et sémantique des objets [74].

Les nouvelles techniques de communications visuelles doivent permettre la capture, la transmission et l'affichage des informations visuelles 3D, puis réaliser l'une des caractéristiques les plus recherchées dans les services de télécommunications de haute qualité, c'est *-la sensation de la réalité 3D-*. Toutefois, les approches stéréoscopiques peuvent fournir la représentation 3D à travers le cerveau humain par la fusion des vues gauches et droites de la même scène, qui sont capturées à partir des angles légèrement différents [79]. Sans aucun doute, ce sera une direction très attrayante et efficace pour réaliser la communication visuelle en 3D dans le futur [75].

Cependant, peu de travaux ont été proposés sur la technique de codage vidéo par fractal [38], [51]. Un système, qui n'est pas vraiment basé-région, a été proposé pour le codage de la vidéo [39], et il est basé sur le partitionnement en quadtree [37].

En 2002, [17] ont proposé un codec fractal d'image basé-région. Cette méthode a permis de nouvelles fonctionnalités au niveau du décodeur, comme la transmission/décodage indépendants de chaque objet dans la vidéo, la recherche basée-région de la vidéo, et surtout l'amélioration de la qualité visuelle du codage de la vidéo [90]. On peut également bénéficier des avantages inhérents du codage fractal, comme le taux de compression et la taille évolutive des sorties pour obtenir un taux évolutif des séquences vidéo décodées, etc. Puisque le processus de décodage est formulé en matière de partitions géométriques et non pas en matière de pixels ou blocs de taille fixe, il est possible de le représenter à n'importe quelle résolution [16]. Toutes ces applications potentielles encouragent le développement d'une nouvelle méthode de codage fractal basé-région de séquences vidéo. Notre contribution est l'utilisation de la DCT adaptée aux objets (Shape-Adaptive

DCT) dans un codeur fractal pour la compression de la vidéo et la stéréo. Les techniques [18, 90] proposent la même idée du basée-région, mais dans le domaine spatial (pixel) et pour la vidéo uniquement. On trouve que l'utilisation du domaine transformé (SA-DCT) peut-être une contribution puisqu'on offre un système de codage permettant une fonctionnalité basée-région sans une pénalité de performance importante (et pour certaines vidéos, même un gain de performance).

Ce chapitre est organisé comme suit : dans la section 3.2, on a brièvement examiné la théorie des systèmes de fonctions itérées dans le domaine transformé. Dans la section 3.3, les méthodes du codage fractal basées-région dans le domaine transformé sont décrites. Les méthodes proposées pour le codage fractal basées-région de séquences vidéo et stéréo sont présentés dans la section 3.4 et la section 3.5, respectivement. Dans la section 3.6 de nombreux résultats expérimentaux sont présentés.

3.2 RE-FORMULATION DE LA THÉORIE DES IFS DANS LE DOMAINE TRANSFORMÉ

Pour améliorer la performance du codage (Sec.2.2.4.3), un schéma hybride entre le codage fractal et la transformation cosinus discrète (DCT) [85, 14, 33] ou bien le codage fractal et les coefficients d'ondelettes de partitionnement adaptative [16, 15, 69] ont été proposés. On présente au-dessous une approche hybride de codage fractal/DCT. Cette approche servira comme base pour la méthode proposée dans ce mémoire.

Dans les méthodes basées sur la DCT, d'abord tous les blocs sources et cibles sont transformés par la transformation $DCT : \hat{I}_{r_i} = DCT(I_{r_i})$ et $\hat{I}_{d_j} = DCT(I_{d_j})$. Soient $\mathbf{u} = (u, v)$ la position d'un coefficient de la DCT 2-D et ζ_i^p les 8 transformations isométriques semblables à celles discutées précédemment (des réflexions spatiales et des rotations) mais adaptées au domaine de la DCT [23], [85].

La transformation contractante σ doit être redéfinie pour représenter les propriétés de la fréquence \mathbf{u} . Plutôt qu'on effectue le sous-échantillonnage spectral, la transformation σ peut être, par exemple, défini comme suit : le $K \cdot Q \times K \cdot Q$ bloc domaine \hat{I}_{d_j} est amené au $Q \times Q$ bloc range \hat{I}_{r_i} en conservant seulement la partie basse fréquence du spectre de \hat{I}_{d_j} [85]. La transformation affine locale w_i est alors calculée en minimisant la mesure de dissimilitude suivante :

$$\varepsilon(\hat{I}_{r_i}, \hat{I}_{d_j}, w_i) = \frac{1}{Card(v_{r_i})} \sum_{\mathbf{u} \in v_{r_i}, \mathbf{u} \neq 0} [\hat{I}_{r_i}(\mathbf{u}) - w_i(\hat{I}_{d_j}, \mathbf{u})]^2, \quad (3.1)$$

Où l'addition est effectuée sur toutes les fréquences de la transformation DCT du bloc range sauf $\mathbf{u} = 0$. Ci-dessus, v_{r_i} dénote l'ensemble des fréquences (Après l'application de la DCT) pour le bloc r_i ; v_{r_i} coïncide avec r_i , mais ces paramètres sont des fréquences discrètes. Comme il a été dit précédemment, l'exclusion de $\mathbf{u} = 0$ est due au fait que l'intensité moyenne du bloc cible r_i (égal à $\hat{I}_{r_i}(0)$) est transmise au lieu du facteur de décalage o_i .

Pour une transformation simplifiée de l'équation (éq.2.11), l'indice j du bloc source, le facteur isométrique ζ_i^p et le facteur d'échelle s_i sont calculés en minimisant ε ci-dessus. Ainsi, pour coder le bloc r_i , pour chaque combinaison de j et ζ_i^p , le facteur d'échelle est calculé comme suit :

$$s_i = \frac{\sum_{\mathbf{u} \in v_{r_i}, \mathbf{u} \neq 0} \sigma(\hat{I}_{d_j}, \zeta_i^p(\mathbf{u})) \hat{I}_{r_i}(\mathbf{u})}{\sum_{\mathbf{u} \in v_{r_i}, \mathbf{u} \neq 0} [\sigma(\hat{I}_{d_j}, \zeta_i^p(\mathbf{u}))]^2},$$

La meilleure combinaison des trois paramètres est celle qui permet d'avoir un ε faible.

3.3 CODAGE FRACTAL BASÉ-RÉGION DES IMAGES DANS LE DOMAINE TRANSFORMÉ

Dans le codage fractal classique (Sec.3.2), le bloc cible est défini indépendamment du contenu de l'image. Les blocs cibles de frontières contiennent des pixels d'un objet et d'autres de l'arrière-plan ou bien d'un autre objet qui se trouve à côté (Fig.2.22). Ainsi, le décodage indépendant d'objets n'est pas possible. Aussi, la qualité de codage peut souffrir de pixels de différentes caractéristiques sur les différents côtés de la frontière ; l'application de la DCT à un tel bloc, mélangera les propriétés spectrales de ces pixels et rendra la recherche d'une correspondance entre le bloc cible et le bloc source peu fiable. Particulièrement une transition d'intensité causera des oscillations spectrales très importantes. Pour contourner ce problème, on propose une nouvelle approche qui prend en compte une segmentation a priori de l'image en région pour coder chaque région indépendamment des autres. Ceci permet une transmission et un stockage indépendants et, donc un décodage indépendant des régions, qui permettra une nouvelle fonctionnalité.

3.3.1 Approche Proposée

Pour contourner les problèmes déjà cités plus haut, on propose d'appliquer la DCT adaptative aux objets (SA-DCT) [71] à chaque segment \mathcal{S} du bloc de frontière des blocs cibles et des blocs sources. Le concept de base de la SA-DCT est d'effectuer d'abord la DCT verticale 1-D sur les pixels actifs et ensuite d'appliquer la DCT horizontale aux coefficients de la DCT verticale avec le même indice de fréquence. La figure (Fig.3.1) illustre la procédure. Les coefficients finaux du SA-DCT sont placés dans le coin supérieur gauche de chaque bloc. Le nombre des coefficients de la SA-DCT est identique au nombre de pixels actifs. Après, la forme de chaque segment est transmise, le décodeur peut exécuter la SA-DCT inverse. L'avantage le plus important de la SA-DCT est sa capacité de s'adapter aux formes arbitraires des régions. Malgré le manque de théorie rigoureuse qui justifie la SA-DCT, sa performance est étonnante et elle est comparable aux méthodes avancées basées sur l'orthogonalisation [41]. De plus, la SA-DCT peut être mise en œuvre en temps réel tandis que les approches basées sur l'orthogonalisation sont coûteuses en temps CPU et en espace mémoire.

En raison de ses propriétés, l'algorithme SA-DCT est devenu un outil commun pour le codage des régions d'image arbitrairement formée et, particulièrement il a été incorporé dans MPEG-4. Dans ce mémoire, on utilise une variante de la SA-DCT, appelé Δ DC-SA-DCT [49]. Elle améliore la performance du SA-DCT par le moyen de deux étapes de traitement supplémentaires : extraction du composant DC du segment \mathcal{S} avant l'application de la SA-DCT et Δ DC correction effectuée au décodage pendant la SA-DCT inverse.

La norme du codage fractal/DCT décrite dans la Section (Sec.3.2) s'applique facilement. En plus, chaque bloc frontière cible est associé à un bloc frontière source du même

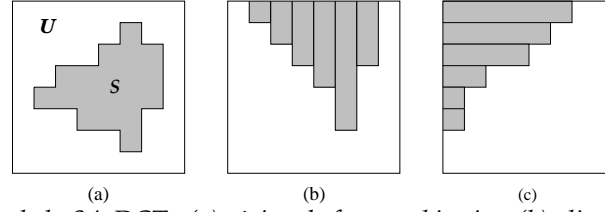


Figure 3.1 – Illustration de la SA-DCT : (a) région de forme arbitraire ; (b) alignement vertical effectué par la DCT verticale 1-D ; (c) alignement horizontal effectué par la DCT horizontale 1-D.

objet. Constamment, cette restriction permet une transmission région-par-région. Chaque segment du bloc cible r_i est traité indépendamment. D'abord, on applique la SA-DCT pour chaque segment $S_{r_i}^m$ du bloc r_i (Fig.3.2), ensuite on trouve les paramètres de la transformation w_i^m pour chaque segment.

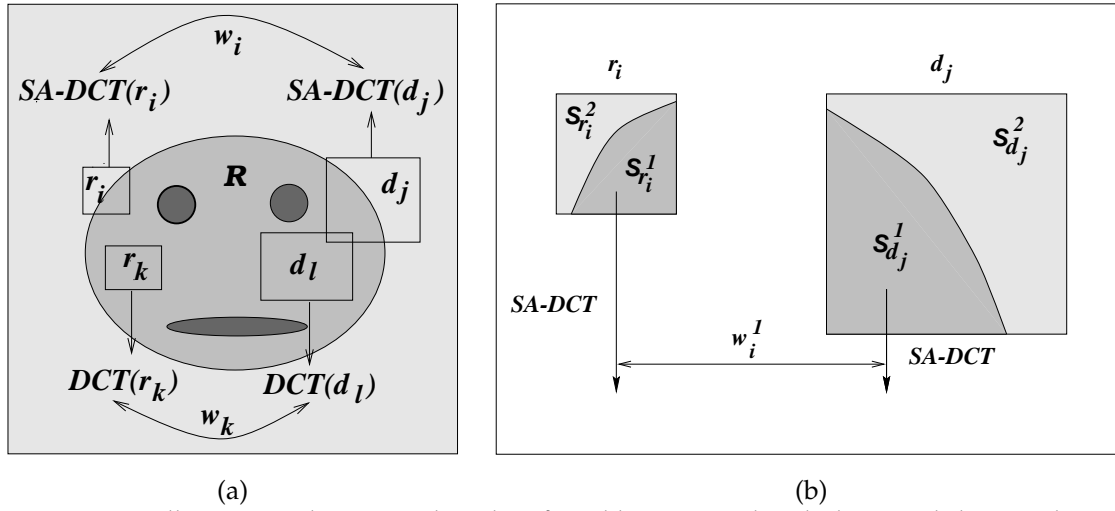


Figure 3.2 – Illustration schématique du codage fractal basé-région dans le domaine de la DCT du système proposé. (a) Les blocs cibles intérieurs sont associés aux blocs sources intérieurs de la même région tandis que les blocs frontières cibles sont partiellement correspondus (éq.3.2) aux blocs frontières sources de la même région. (b) Les segments de blocs frontières sont traités indépendamment : le segment $S_{r_i}^1$ est calculé à partir d'un segment du bloc source d_j par une transformation affine w_i^1 , Par contre le segment $S_{r_i}^2$ du même bloc cible r_i est calculé à partir d'un autre bloc source par une transformation différente. Les blocs intérieurs sont transformés par la DCT, tandis que les segments des blocs frontières sont transformés par l'utilisation de la SA-DCT

Rappelons que \hat{I}_{r_i} et \hat{I}_{d_j} sont les transformations DCT des intensités des blocs I_{r_i} et I_{d_j} . Posons $\mathcal{P}_{r_i}^m$ (Fig.3.3) le segment $S_{r_i}^m$ (m -ème segment frontière du bloc cible r_i) après l'application de la SA-DCT. Notez que la forme de $\mathcal{P}_{r_i}^m$ diffère de celle de $S_{r_i}^m$ en raison des décalages verticaux et horizontaux, mais le nombre de pixels est inchangé. En plus, $\hat{I}_{r_i}^m(\mathbf{u})$ est un coefficient SA-DCT dans $\mathcal{P}_{r_i}^m$ pour la fréquence \mathbf{u} . Pour trouver le meilleur bloc source d_j pour un bloc cible donné r_i , la mesure de dissimilitude suivante dans le domaine transformé est utilisée :

$$\varepsilon = \frac{1}{\text{Card}(\mathcal{P}_{r_i}^m)} \sum_{\mathbf{u} \in \mathcal{P}_{r_i}^m, \mathbf{u} \neq 0} [\hat{I}_{r_i}^m(\mathbf{u}) - w_i^m(\hat{I}_{d_j}^m(\mathbf{u}))]^2, \quad (3.2)$$

Où $\tilde{I}_{d_j}^n$ représente l'extrapolation du n-ème segment d'intensité du bloc source transformé par la SA-DCT :

$$\tilde{I}_{d_j}^n(\mathbf{u}) = \begin{cases} \hat{I}_{d_j}^n(\mathbf{u}) & \text{Si } \mathbf{u} \in \mathcal{P}_{d_j}^n, \\ v & \text{Autrement.} \end{cases} \quad (3.3)$$

Bien que diverses valeurs v puissent être utilisées, un choix logique, qu'on adopte ici, est de mettre v à zéro (le ZÉRO padding) puisque tous les coefficients sont en hautes fréquences.

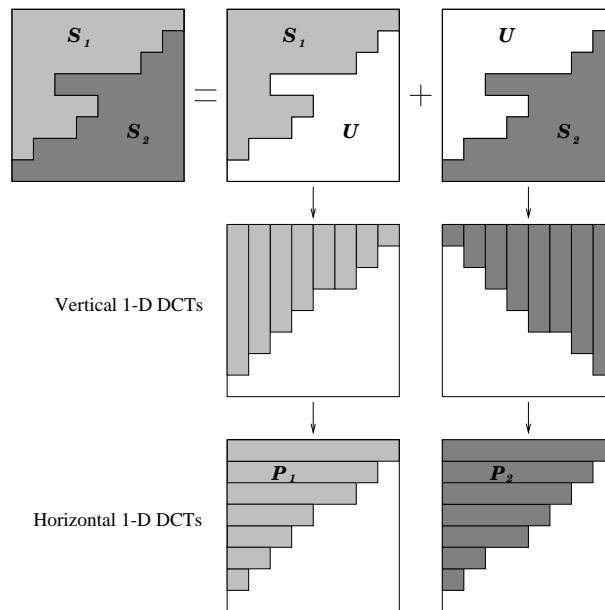


Figure 3.3 – Exemple d'application de la SA-DCT pour un bloc frontières avec deux segments. Notons que dû aux décalages horizontaux et verticaux, la forme de \mathcal{P}_i est différente de celle de S_i .

Clairement, les contributions en ε sont seulement faites aux indices des segments de bloc cible $\mathcal{P}_{r_i}^m$. Si pour la fréquence \mathbf{u} , un coefficient dans $\mathcal{P}_{r_i}^m$ existe tandis qu'il n'existe pas dans le segment $\mathcal{P}_{d_j}^n$ du bloc source en considération, alors une contribution significative en ε sera faite. Ainsi, une correspondance devrait être effectuée entre les segments des blocs cibles et des blocs sources qui ont des formes semblables ou bien une partie du contenu spectral des coefficients de bloc cible qui sont similaires sans avoir forcément une correspondance avec le segment du bloc source. Dans le cas où un segment du bloc source $\mathcal{P}_{d_j}^n$ englobe entièrement $\mathcal{P}_{r_i}^m$, il va probablement être accepté si les coefficients de même indice sont semblables dans $\mathcal{P}_{d_j}^n$ et dans $\mathcal{P}_{r_i}^m$. Malheureusement, si les formes ou les tailles des segments sont très différentes, les fonctions de base pour $\mathcal{P}_{d_j}^n$ et $\mathcal{P}_{r_i}^m$ peuvent différer considérablement; malgré la similarité des valeurs de coefficient, les modèles d'intensité dans le domaine spatial peuvent être tout à fait différents dans les deux cas ainsi une dégradation de performances.

3.3.2 Choix de l'espace de recherche

Pour assurer un codage/décodage région-par-région, les blocs sources et cibles doivent être localisés dans la même région \mathcal{R} . Donc, pour minimiser la distorsion (éq.3.2), quatre cas pour l'emplacement de blocs r_i et d_j , en ce qui concerne la région \mathcal{R} , peuvent surgir :

1. r_i et d_j sont des blocs intérieurs (int/int),
2. r_i et d_j sont des blocs frontières (fron/fron),
3. r_i est un bloc intérieur tandis que d_j est un bloc frontière (int/fron),
4. r_i est un bloc frontière tandis que d_j est un bloc intérieur (fron/int).

Dans le premier cas, la recherche standard est exécutée sur les blocs intérieurs de la région (éq.3.1), dans le deuxième cas seulement la correspondance partielle est effectuée utilisant la mesure de dissimilitude (éq.3.2) tant que les blocs cibles et sources sont sur la frontière de la région. Clairement, le remplissage d'intensité (éq.3.3), est nécessaire quand $\mathcal{P}_{d_j}^n$ n'englobe pas entièrement $\mathcal{P}_{r_i}^m$.

Dans le troisième cas, l'extrapolation d'intensité (éq.3.3) du bloc source est toujours nécessaire tandis que r_i est un bloc intérieur et d_j est un bloc de frontière. Bien que d'habitude une meilleure solution que l'utilisation d'intensités de la région voisine (le codage fractal standard), le remplissage est probablement le résultat des intensités sous-optimales (le remplissage imprécis) en comparaison avec des blocs sources entièrement englobés dans la région.

Dans le quatrième cas, aucun remplissage n'est nécessaire ; r_i est un bloc frontière tandis que d_j est un bloc intérieur. Bien que ceci soit un scénario faisable, ce cas exige un examen coûteux de tous les blocs sources intérieurs pour chaque bloc cible frontière ; typiquement il y a beaucoup plus de blocs intérieurs que des blocs frontières.

En sautant le troisième et le quatrième cas, la recherche 1 et 2 permet d'éliminer les calculs supplémentaires par la recherche des blocs frontières seulement entre eux et la recherche des blocs intérieurs seulement dans les blocs intérieurs. La figure (Fig.3.2 (a)) illustre une représentation schématique de la recherche 1 et 2.

Comme indiqué dans la figure (Fig.3.2(b)) la recherche de paramètres de la transformation w_i est effectué indépendamment pour chaque segment du bloc cible. Cependant, une réduction significative de la complexité de calcul est possible si la position des blocs sources communs j (**b**) et l'isométrie ζ_i^p sont utilisés pour tous les segments, mais s_i et m_{r_i} sont calculés indépendamment pour chaque segment pour permettre la correspondance précise de textures. On a expérimenté avec cette approche et on a observé une dégradation de performance de moins que 0.1dB en atteignant 20-30% de réduction du temps CPU de codage.

3.3.3 Encodage des paramètres

Les paramètres suivants doivent être codés pour la transmission ou le stockage : le facteur d'échelle photométrique s_i et la moyenne d'intensités du bloc cible m_{r_i} , le vecteur de translation \mathbf{b} (exprimé en matière de la position relative de d_j en ce qui concerne r_i) et la transformation affine \mathbf{A} réduite à 4 rotations et 4 réflexions.

Puisque s_i et m_{r_i} ont, en général, des distributions non uniformes, le codage entropique est avantageux. On a choisi la quantification sur 5 bits de s_i et la quantification sur 7 bits de m_{r_i} , prouvé dans [37] pour leur bonne performance, puis on l'a suivi avec le codage de Huffman.

Pour s'assurer que l'encodeur et le décodeur utilisent les mêmes valeurs s_i et m_{r_i} , on les quantifie pendant la minimisation de la mesure de dissimilitude (éq.3.2), c'est-à-dire, avant chaque évaluation de ε . L'utilisation des valeurs flottantes dans la minimisation et leurs quantifications seulement avant la transmission dégradent la performance. On code le vecteur \mathbf{b} qui représente la position relative de d_j par rapport à r_i par des mots de code de longueur fixe (déterminé par la taille de l'image). On utilise des mots de code de 3 bits pour coder les 8 rotations/réflexions possibles.

Les modèles de segmentation (les alphas-planes) doivent aussi être transmis afin d'extraire les segments appropriés S^m du bloc source pendant le décodage et pour permettre une application correcte des transformations w_i^m . Le lecteur est renvoyé à la littérature (par exemple, [62]) pour le codage des alphas-planes. Notons que dans l'état de l'art, il existe des méthodes sans perte permettant la transmission typique des modèles de segmentation (le test de séquences dans le MPEG-4) pour seulement 0.01-0.03bpp [53].

3.4 CODAGE FRACTAL BASÉ-RÉGION DES SÉQUENCES VIDÉO

[89, 90] ont proposé un nouvel algorithme basé-région pour le codage fractal des séquences vidéo, qui est basé sur le système hybride CPM/NCIM [51], [52] présenté dans le chapitre précédent. Ce nouvel algorithme fournit la fonctionnalité basée-région pour le codage de la vidéo monoculaire basé sur son alpha-plane (Sec.3.3).

Le codage hybride CPM/NCIM combiné avec le codage fractal de la vidéo et les algorithmes d'estimation et de compensation de mouvement (EM/CM), exploite la corrélation temporelle élevée entre les trames adjacentes. Dans les deux algorithmes CPM et NCIM [52, 51], chaque bloc cible est compensé par un bloc source de la trame précédente, qui a la même taille que le bloc cible, alors que le bloc source était toujours plus grand que le bloc cible dans les codeurs d'images fractals conventionnels (Sec.3.2).

La différence principale entre le CPM et le NCIM est que le CPM devrait être contractant, pour que le processus itératif de décodage puisse converger tandis que le NCIM n'a pas besoin d'être contractant puisque le décodage dépend des trames déjà décodées et il est non itératif. Les quatre premières trames de la séquence vidéo sont traitées comme un groupe de codage et sont codées en appliquant le CPM, chaque trame est codée à partir des blocs des quatre trames circulaires précédentes. Autrement dit, K^{th} la trame de F_k est divisée en blocs cibles et chaque bloc cible r_i dans F_k est estimé à partir d'un bloc source d_j dans F_{k-1} (Fig.2.16).

La structure du NCIM est la même que les transformations inter-trames, qui forme le CPM, sauf qu'il n'y a aucune contrainte sur les coefficients de contraste s_i . Puisque le déplacement des séquences d'images entraîne des corrélations temporelles élevées, les transformations cibles-sources deviennent plus efficaces si la taille du bloc source d_j est même que celle du bloc cible r_i (Voir la Figure 2.21(b)). Dans ce cas, la transformation cible-source peut être interprétée comme une technique d'estimation et de compensation de mouvement.

Dans ce contexte, l'avantage principal du modèle de transformation cible-source proposé consiste en mouvement réel dans les séquences d'images, des petits vecteurs de mouvement sont plus probables que de plus grands. Donc, la région de recherche pour les vecteurs de mouvement peut être localisée dans la zone près de l'emplacement du bloc cible [91, 20]. Dans le décodeur, les quatre premières trames sont reconstruites en appliquant itérativement le CPM. Alors les trames restantes peuvent être reconstruites en appliquant le NCIM aux trames reconstruites précédemment par le CPM sans exiger l'itération, puisque le NCIM n'est pas un modèle contractant. Les quatre premières trames codées par le CPM sont l'ensemble minimal pour décoder toutes les trames [32] et ils peuvent être décodés sans références à d'autres trames. Donc, seulement le CPM affecte la convergence du modèle fractal total et c'est la raison pour laquelle le NCIM n'a pas besoin d'être contractant.

De plus, on utilise un modèle d'estimation/compensation de mouvement basé sur le partitionnement en quadtree. Un quadtree décrit un objet en plaçant les blocs qui ne chevauchent pas de tailles différentes à l'intérieur de l'objet pour que l'objet puisse être décrit aussi précis que possible. Typiquement les tailles des carrés sont des puissances de 2. On fixe la taille carrée minimale du quadtree pour être 4x4 pixels.

Dans une première étape, on place les carrés de taille maximale $M \times M$ à côté les uns des autres sur l'image entière. Après on décide si on doit diviser un carré en se basant sur le critère d'homogénéité [68], comme indiqué dans la Figure (Fig.2.21(a)). Si oui, donc on remplace le carré par des sous-carrés de taille $M/2 \times M/2$ pixels. On répète ce processus de façon itérative jusqu'à ce qu'on atteigne une taille minimale du bloc cible (4x4 pixels) ou bien une distorsion minimale ε (éq.3.1), qui est fixée par les utilisateurs conformément aux exigences de la qualité vidéo. Pour fournir la fonctionnalité basée-région, la séquence vidéo sera codée bloc par bloc et région par région selon son alpha-plan. En utilisant des blocs cibles et sources rectangulaires, on divise les blocs frontières en segments appartenant aux objets différents comme illustré dans la figure (Fig.2.21(b))

3.5 CODAGE FRACTAL BASÉ-RÉGION DES SÉQUENCES STÉRÉO

3.5.1 Codage basé-bloc et le profile multi-vue de MPEG2

La plupart des techniques pour la compression de séquences stéréo sont à la base du codage basé-blocs des séquences stéréo défini dans le profile multi-vue de MPEG-2 [7]. Avec cette approche, par exemple, le codeur compresse premièrement la vue droite (vue de base) avec l’algorithme de codage des séquences vidéo monoculaires. Pour coder la vue gauche (la vue d’amélioration), chaque macro-bloc est prédéfinie à partir de la vue droite en utilisant l’algorithme prédictif de compensation de disparité (Disparity Compensated Prediction, (DCP)), et à partir de la trame précédente de la vue gauche en utilisant l’algorithme prédictif de compensation de mouvement (Mouvement Compensated Prediction, (MCP)) comme indiqué dans la figure (Fig.3.4(a)). Les deux algorithmes sont utilisés et l’erreur de prédiction est alors codée selon celui qui donne l’erreur la plus petite.

Pour se servir des codeurs existants pour le mono-séquence, les vecteurs de disparité peuvent être estimés par la même façon que l’estimation de mouvement, c’est-à-dire, supposant la disparité des blocs constants et trouvant la meilleure macro-bloc de correspondance dans la vue droite [75]. L’inconvénient de cette approche est que les disparités évaluées pour des macro-blocs sont d’habitude discontinues et ont des artefacts visibles à cause de la taille fixe des blocs même avec la division de bloc adaptatif.

Dans [78], les vues d’entrées sont d’abord alignées pour faire leurs lignes horizontales

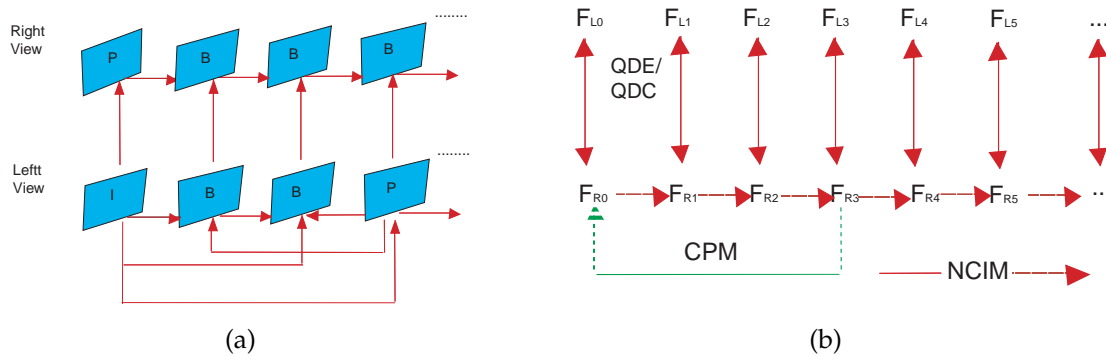


Figure 3.4 – illustration schématique de la prédiction inter-vue. (a) Le profile multi-vue de MPEG-2 [7] (b) Schéma hybride de codage de séquence stéréo en combinant CPM / NCIM avec QDE / QDC.

et l’estimation de la disparité est alors appliquée aux images alignées. Quelques régions deviennent déformées parce que la prédiction de la vue gauche à partir de la vue droite originale utilise un bloc de taille fixe. En plus, des artefacts visibles apparaissent dans l’algorithme de recherche de blocs, mais il garde la vue prédictive fidèle que la vue originale puisque le PSNR est plus élevé.

L’algorithme utilisé dans [78] adopte des blocs de tailles fixes de 16x16 pixels et une zone de recherche de 100 pixels. Mais, si on adopte des blocs de taille variable voir (Fig.2.21(a)) et on augmente la zone de recherche, on peut obtenir non seulement un PSNR élevé, mais

aussi l'image prédite sera visiblement plus net. Dans [65, 64], pour synthétiser une image 3D ayant un plus grand nombre de vues qu'une image 3D ayant moins de vues, une nouvelle technique hybride d'interpolation et de compression fractale s'adaptant à n'importe quel nombre de vues a été proposée. Mais cette méthode est seulement applicable pour des images multi-vues basées sur l'interpolation, les images stéréo ne sont pas suffisantes pour utiliser cette technique.

3.5.2 Nouvelle méthode de codage fractal des séquences stéréo

Essentiellement, la susdite technique de codage basé-bloc rapproche les formes d'objets dans la scène avec des blocs carrés de taille fixe. Donc, ce codeur génère des erreurs de prédiction plus importantes dans les blocs frontières d'objets. Ces blocs frontières contiennent deux (ou plus) d'objets avec des mouvements différents, qui ne peuvent pas être représentés par un vecteur de mouvement simple. Des codeurs dépendants reconnaissent de tels problèmes [67] et essaient de segmenter une trame vidéo dans des régions correspondant aux objets différents et de coder ces objets séparément. Pour chaque objet, en plus des informations de texture et de mouvement, les informations de région doivent être transmises. Dans [81], un encodeur stéréoscopique basé sur MPEG-4 a été proposé, dans lequel la vue principale est codée en utilisant l'encodeur MPEG-4 et les vues auxiliaires sont codées selon le mouvement commun et la compensation de disparité.

Sur la base de notre codeur fractal basé-région de séquence vidéo monoculaire (Sec 3.4), on code la vue droite (la vue de base) de séquence vidéo par le codeur basé-région CPM/NCIM basé sur le partitionnement en quadtree et on désigne la séquence vidéo de la vue droite reconstruite pendant le processus de codage comme la référence pour prévoir la vue gauche (la vue d'amélioration) de la séquence stéréo. On propose d'utiliser les transformations affines et l'estimation/compensation de disparité basée sur le partitionnement en quadtree, ou d'appliquer simplement l'estimation/compensation de mouvement CPM/NCIM sur la vue gauche de la séquence vidéo. Parce que les tailles des blocs sont variables et les blocs cibles et sources sont pré-classées [37], on peut demeurer que les modèles basés sur le quadtree puissent fournir de bonnes performances pour le PSNR et la qualité visuelle avec beaucoup moins de temps de calcul que la recherche exhaustive et les méthodes itératives.

Un avantage du codage fractal qui utilise le CPM/NCIM sur la séquence vidéo de vue droite, est qu'il fournit des taux de compression plus élevés que d'autres méthodes de compression. Après la transformation affine et l'estimation de disparité basée sur le quadtree entre les trames de vues gauches et droites, on peut seulement enregistrer les positions ou les paramètres des transformations affines complètes des blocs similaires de la trame gauche relative aux trames des vues droites reconstruites.

La trame gauche prédite peut être obtenue en remplaçant chaque bloc par le meilleur bloc correspondant dans la trame droite de référence, ainsi on peut arriver à un taux de

compression beaucoup plus élevée pour la séquence vidéo de vue gauche et pour la vue droite de référence. Aussi, on peut juste appliquer le CPM/NCIM sur la séquence vidéo de vue gauche, cela dépend du contenu de la vidéo et des demandes d'utilisateurs sur le débit binaire et le PSNR.

3.5.2.1 Schéma du codeur fractal des séquences stéréo

En combinant les techniques qui ont été présentées plus haut, on peut synthétiser la procédure du schéma de codage comme suit :

1. Codage de la vue droite de séquence vidéo par l'algorithme d'estimation/compensation de mouvement basé-région CPM/NCIM basé sur le partitionnement en quadtree (Sec3.4), ainsi on peut exploiter les corrélations temporelles élevées entre les trames adjacentes ;
2. Décodage de la vue droite de séquence vidéo en utilisant l'algorithme de décodage basé-région CPM/NCIM basé sur le partitionnement en quadtree ;
3. Codage de la vue gauche de séquence vidéo en exploitant les corrélations spatiales entre les séquences vidéo gauches et droites par l'estimation de disparité basé sur le quadtree (QDE Quadtree Disparity Estimation). Notez que si les contextes sont très différents, CPM/NCIM sera appliqué pour la vue gauche au lieu de QDE ;
4. Décodage de la vue gauche de séquence vidéo par la compensation de disparité basée sur le quadtree (QDC Quadtree Disparity Compensation). L'algorithme de codage basé-région CPM/NCIM basé sur le partitionnement en quadtree pour la vue gauche peut être utilisé selon le processus de codage.

Le système de codage des séquences stéréo proposé, qui combine l'algorithme d'estimation/compensation de mouvement (CPM/NCIM) avec l'algorithme d'estimation /compensation de disparité (QDE/QDC), est illustré dans la figure (Fig.3.4(b)).

3.6 RÉSULTATS EXPÉRIMENTAUX

Dans cette section, on présentera les résultats obtenus suite à l'implémentation du codec hybride basé-région CPM/NCIM et basé sur le partitionnement en quadtree dans le domaine transformé. Les résultats démontrent que des débits très faibles peuvent être atteints avec un taux de compression acceptable. On évalue l'effet des niveaux minimum et maximum du partitionnement sur le codage, puis une comparaison analytique entre le codage dans le domaine transformé et celui du domaine spatial est faite pour justifier l'importance d'employer les transformations (DCT et SA-DCT) dans le processus du codage.

3.6.1 Codage fractal basé-région de séquence vidéo

Pour évaluer la performance du codec proposé pour les séquences vidéo monoculaires, on utilise trois séquences du type CIF de dimensions 352x288 pixels et une séquence du type SIF, de dimension 352x240 pixels. Pour les tests on a choisi quatre séquences vidéo monoculaires : *Childrens* (352x288 pixels, 80 trames, 8.33 trames/seconde), *News* (352x288 pixels, 24 trames, 8.33 trames/seconde), *Weather* (352x288 pixels, 24 trames, 8.33 trames/seconde) et *Cyclamen* (352x240 pixels, 20 trames, 8.33 trames/seconde) et leurs alphas-plans.

L'expérience est déroulée dans un PC sur une machine Intel (R) Xeon (R) Quad-Core X5650 (2.67 GHz and 2.66 GHz, deux processeurs, 12 GB RAM). Les algorithmes sont mis en œuvre sur un système d'exploitation Linux Fedora 9.0.

La valeur suprême des algorithmes proposés peut seulement être évaluée dans le contexte de trames complètes composées de plusieurs objets. Donc, on a codé premièrement les quatre séquences vidéo monoculaires comme des séquences complètes sans fonctionnalité basée-région (Section 3.2). Deuxièmement, on les code objet-par-objet en utilisant le schéma de codage basé-région proposé (Sec.3.3) et (Sec.3.4). Le tableau (Tab.3.1), contient les valeurs des paramètres fractals et d'implémentations utilisées pour les différents tests.

Paramètres	valeur
Nombre de Bits pour coder s	5
Nombre de Bits pour coder o	7
Nombre de trame pour CPM	4
Taille de la fenêtre de recherche	32
Nombre d'itération pour le décodage CPM	10
Minimum du partitionnement quadtree	5
Maximum du partitionnement quadtree	7

Table 3.1 – Paramètres fractals et d'implémentations pour le système proposé

On étudie maintenant, l'effet des niveaux minimum et maximum du partitionnement quadtree sur le codage basé-région dans le domaine transformé. Les tableaux (Tab.3.2, Tab.3.3) contiennent les résultats de la qualité (PSNR), de la largeur de bande moyenne pour transférer la séquence (BW) et du temps de codage pour les séquences *News* et *Weather* respectivement codés en 20 trames. Les paramètres variables sont Min-Part et Max-Part qui représentent le minimum et le maximum du partitionnement en quadtree.

Min-Part	Max-Part	PSNR(r_0)	PSNR(r_1)	PSNR(Plein)	BW(r_0)	BW(r_1)	BW (Plein)	Temps (S)
4	5	25.57	20.32	22.96	55.48	35.92	91.41	9 s
4	6	29.11	22.83	25.82	121.11	105.77	226.89	15 s
4	7	32.72	26.05	29.14	201.93	251.34	462.89	26 s
4	8	35.81	33.86	35.02	323.89	324.01	647.52	1 min 9 s
5	7	33.18	26.78	29.08	245.41	263.45	508.86	24 s
5	8	37.04	33.54	35.48	355.90	340.54	696.44	1 min 16 s
6	8	39.00	34.86	37.07	502.46	420.81	923.27	1 min 16 s

Table 3.2 – Effet de Min-part et Max-part du partitionnement quadtree sur le codage basé-région de la séquence *News-Cif* codée en 20 trames.

Min-Part	Max-Part	PSNR(r_0)	PSNR(r_1)	PSNR(Plein)	BW(r_0)	BW(r_1)	BW (Plein)	Temps (S)
4	5	22.01	15.08	16.15	28.62	63.54	92.17	5 s
4	6	24.86	18.62	19.64	88.13	215.20	303.34	15 s
4	7	29.09	24.42	25.29	212.32	582.92	795.24	52 s
4	8	33.41	34.62	34.25	352.46	270.97	623.43	1 min 25 s
5	7	28.59	24.42	25.23	216.99	594.82	811.81	51 s
5	8	31.16	35.16	33.65	364.21	317.56	681.78	1 min 21 s
6	8	37.21	36.20	36.46	403.52	495.54	899.07	1 min 28 s

Table 3.3 – Effet de Min-part et Max-part du partitionnement quadtree sur le codage basé-région de la séquence *Weather-Cif* codée en 20 trames.

Où, r_0 représente les résultats de la région 0, r_1 représente ceux de la région 1 et *plein* représente ceux de l'ensemble (images pleines).

Dans le non basé-région, plus le partitionnement minimum était élevé, moins de gros blocs cibles étaient codés, la largeur de bande augmentait et améliorait de peu, la qualité. Ici, ce n'est pas le cas. En passant de Min-part=4 et Max-part=7 à 5 et 7, on voit le PSNR de la séquence *News* pleine chuter de 0.06 dB. Ce phénomène s'explique par le fait que le codage de plus grands blocs frontières peut être plus efficace dans certains cas. Mais dans la plupart des cas, plus le partitionnement maximum est élevé, plus les blocs sont petits, la qualité et la largeur de bande augmentent.

Les figures (Fig.3.5(a), Fig.3.5(b)) illustrent la valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande (BW) pour la séquence *News-cif* et la séquence *Weather-cif* respectivement de 20 trames chacune. Chaque point représente une valeur d'exécution en changeant le Min-Part et le Max-Part. Dans la séquence *New-Cif*, pour des largeurs de bandes supérieures à 420Kbps, les résultats sont de meilleure qualité. Par contre pour la séquence *Weather-Cif*, la qualité s'améliore à partir de la largeur de bande 600kbps. On

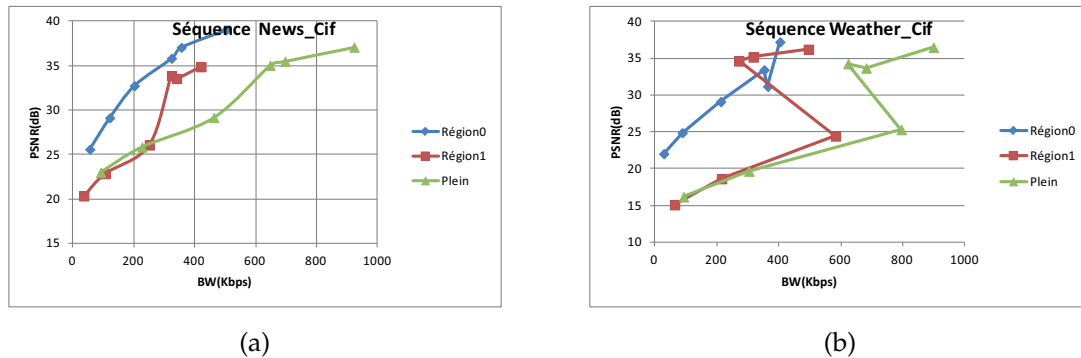


Figure 3.5 – illustration schématique de la valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne pour une séquence de 20 trames(BW). (a) Séquence News-cif(b) Séquence Weather-cif.

remarque pour la figure (Fig.3.5(b)), que pour une largeur de bande du cinquième point comparable avec le quatrième point d'exécution, la qualité chute presque de 0.6dB, ce phénomène s'explique par le fait que le partitionnement utilisé dans le cinquième point n'est pas adéquat pour ce type de séquence. La région 0 pour le New-Cif et la région 0 pour le weather-cif montrent de bons résultats en matière de qualité. Ceci s'explique par le fait que ces régions ont peu de mouvements, donc une corrélation très élevée entre les trames que le NCIM peut corriger les faibles résultats du CPM dans les quelques trames qui suivent.

Dans le même cadre, les figures (Fig.3.6(a), Fig.3.6(b)) illustrent les résultats pour le même type de test appliqué sur les séquences Children-Cif et Cyclamen-Sif respectivement.

Pour les deux séquences Children-Cif et Cyclamen-Sif, les résultats deviennent meilleurs

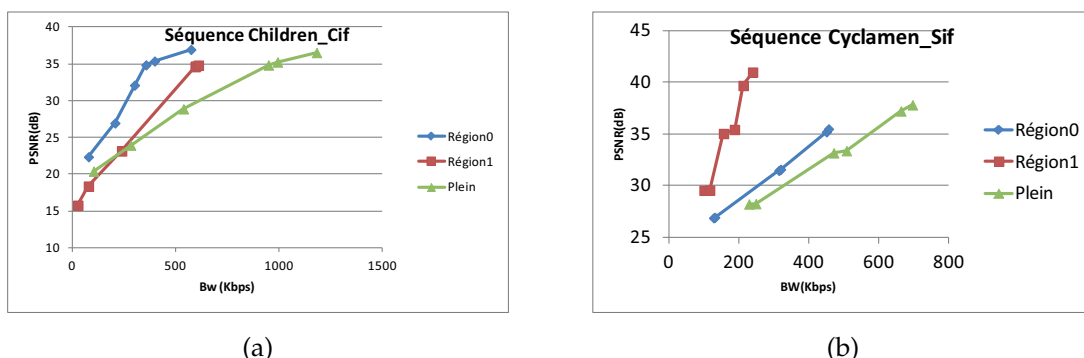
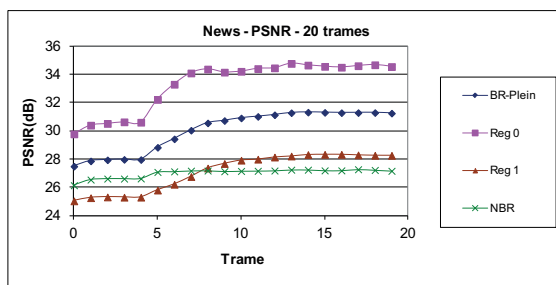


Figure 3.6 – illustration schématique de la valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne pour une séquence de 20 trames(BW). (a) Séquence Children-Cif(b) Séquence Cyclamen-Sif.

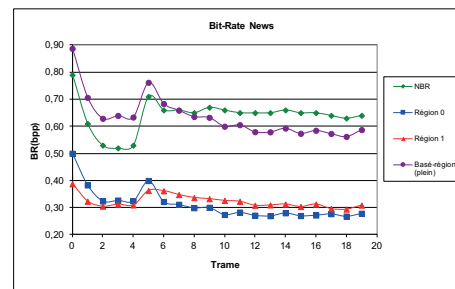
tant que le partitionnement se rend plus profond. La qualité pour la région 0 de la séquence Children-Cif et la région 1 pour la séquence Cyclamen-Sif est bonne, car ces deux régions ont peu de mouvement et le NCIM peut mieux corriger les erreurs du CPM dans les trames qui suivent.

D'après les résultats trouvés, on peut constater que le minimum et le maximum du partitionnement en quadtree influencent d'une façon significative la qualité du codeur ainsi que la largeur de bande nécessaire pour la transmission.

Une comparaison du codage basé-région avec le non basé-région dans le domaine spatial a été faite dans [18]. Ils ont prouvé la performance du codage basé-région en matière de qualité, de largeur de bande et de temps d'exécution. Cette méthode a été exploitée par [91] pour le codage de la stéréo dans le domaine spatial. On montre dans ce qui suit, les résultats de [18] pour les séquences *News* et *Childrens* pour les 20 trames. La figure (Fig.3.7(a)) trace les courbes du PSNR et la figure (Fig.3.7(b)) le Bit-Rate pour les 20 trames de *News*. La figure (Fig.3.8(a)) illustre les courbes du PSNR et la figure (Fig.3.8(b)) le Bit-Rate pour les 20 trames de *Children*. Notez que *Région0* dénote le premier-plan, *Région1* dénote l'arrière-plan, NBR dénote le "non basé-région" et BR-Plein dénote le "basé-région plein" et que la tolérance utilisée pour ces expériences est de 5.

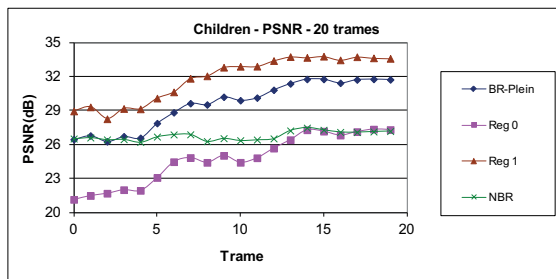


(a) Courbes PSNR séquence News

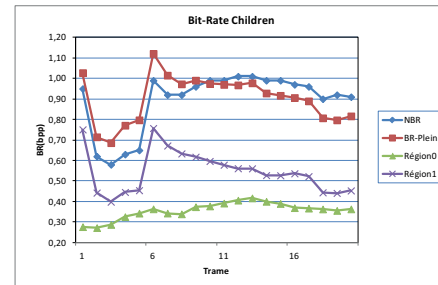


(b) Courbes débit binaire séquence News

Figure 3.7 – Comparaison entre le codage fractal basé-région (BR-Plein) et le codage fractal non basé-région(NBR) dans le domaine spatial pour la séquence News de 20 trames.



(a) Courbes PSNR séquence Children



(b) Courbes débit binaire séquence Children

Figure 3.8 – Comparaison entre le codage fractal basé-région (BR-Plein) et le codage fractal non basé-région(NBR) dans le domaine spatial pour la séquence Children de 20 trames.

D'après les résultats trouvés dans le domaine spatial, le codage fractal basé-région donne de meilleurs résultats en termes de qualité et de la largeur de bande. Il reste à prouver ces résultats dans le domaine fréquentiel puis faire une comparaison entre le codage fractal basé-région et basé sur le partitionnement en quadtree de la vidéo mono-culaire dans le domaine spatial et celui dans le domaine transformé.

Maintenant, on va comparer le codage basé-région de la méthode proposée (Sec.3.3) avec celui de l'image entière (Sec.3.2). On montre les résultats des quatre séquences vidéo

suivantes : *News*, *Children*, *Weather* et *Cyclamen* qui sont constituées de 20 trames chacune. Les figures (Fig.3.9), (Fig.3.10), (Fig.3.12) et (Fig.3.11) illustrent une comparaison du basé-région (Region0 (DCT+SADCT), Region1 (DCT+SADCT), Plein (DCT+SADCT)) avec le non basé-région (NBR(DCT)) en termes du PSNR et du bit rate pour les quatre séquences vidéo précédemment décrites respectivement. La tolérance utilisée pour ces expériences est par défaut 6.5.

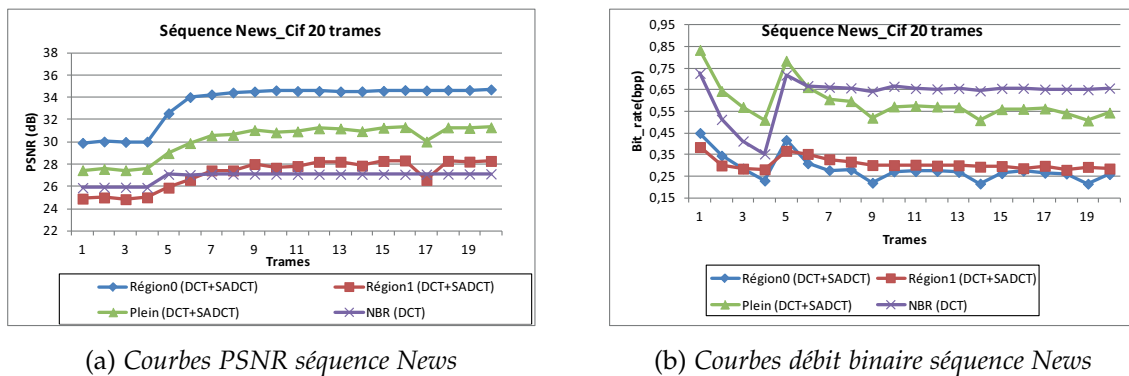


Figure 3.9 – Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence News de 20 trames.

La partie gauche (Fig.3.9(a)) représente la valeur de la qualité (PSNR) pour chaque trame, et la partie droite (Fig.3.9(b)) illustre le taux de bit par pixel pour les 20 trames de la séquence *News*. D'après les deux figures on constate que la qualité (PSNR) du basé-région est élevée par rapport au NBR, cependant le bit-rate s'améliore après la cinquième trame et devient mieux que le NBR. En ce qui concerne les régions, la qualité et le Bit-rate de la région0 sont meilleurs en comparaison avec la Région1, car cette région a peu de mouvements et le NCIM peut facilement corriger les erreurs du CPM.

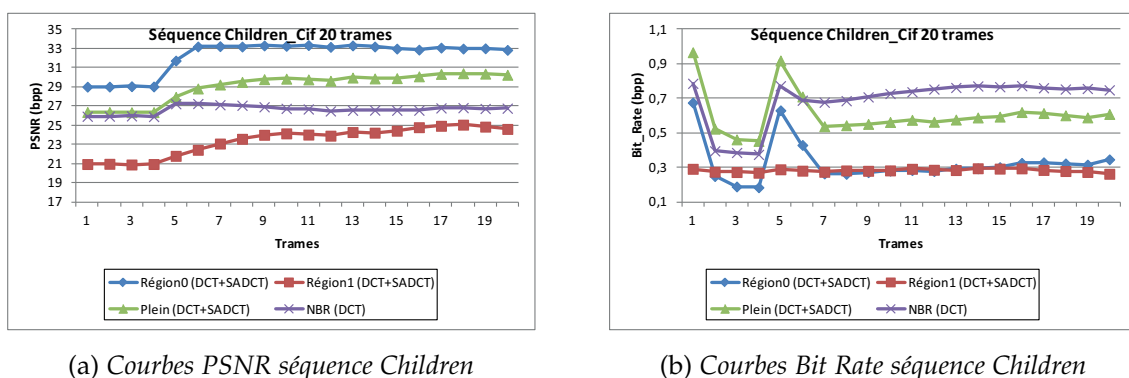


Figure 3.10 – Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence Children de 20 trames.

La figure (Fig.3.10(a)) illustre la qualité (PSNR) en fonction du nombre de trames pour la séquence *Children* de 20 trames. Le NBR a la même qualité que le basé-région pour les quatre premières trames, après que le codage NCIM prend la relève la qualité du

codage fractal basé-région s'améliore d'une manière significative et surpasse la qualité du NBR. Pour le bit rate (Fig.3.10(b)), le codage fractal non basé-région présente un bit-rate réduit par rapport au codage fractal basé-région pour les six premières trames, après le bit rate du codage fractal basé-région diminue et devient plus faible que celui du NBR. En plus, la région 0 présente un mouvement faible par rapport à la région 1 alors son PSNR est élevé et son bit-rate est faible mais il est comparable avec celui de la région 1.

Ce qui a été dit pour la séquence *Children* est applicable pour la séquence *Weather* suivante sauf que l'amélioration du PSNR pour le basé-région est à partir de la trame numéro 5, et pour le bit-rate est à partir de la trame numéro 11. Voir figure (Fig.3.12(a), Fig.3.12(b)).

Passant maintenant à la séquence *Cyclamen*. Les figures (Fig.3.11(a),Fig.3.11(b)) montrent

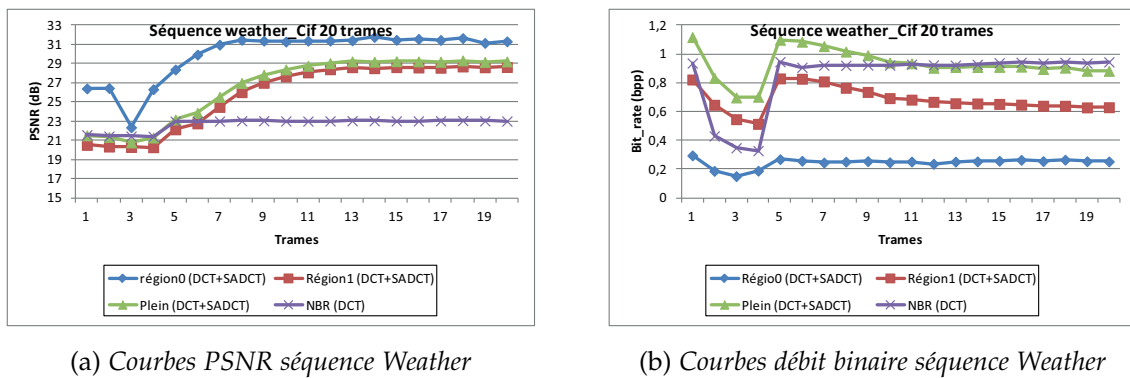


Figure 3.11 – Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence Weather de 20 trames.

une amélioration du PSNR et du bit-rate du NBR par rapport au basé-région. Ceci revient à la nature de cette séquence qui présente beaucoup de blocs frontières avec un mouvement très rapide entre les trames, ce qui rend le codage basé-région difficile. Cependant, le codage non basé-région devient bénéfique pour ce type de séquence.

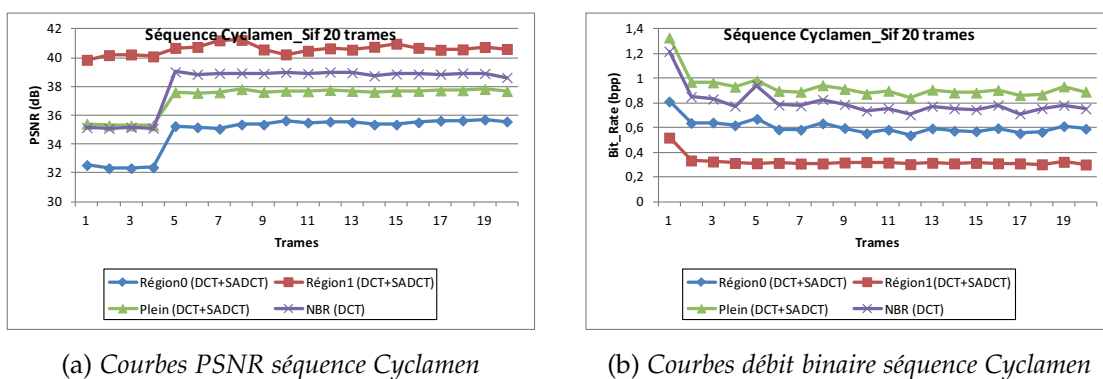


Figure 3.12 – Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence Cyclamen de 20 trames.

Après que le codage NCIM prend la relève du codage CPM, la qualité augmente

graduellement pour atteindre un niveau de qualité supérieur. Dans le codage basé-région, la corrélation temporelle pour une région fixe, comme l'arrière-plan, est très élevée. Le codage NCIM dans ces régions corrige les effets de contrainte de 1.0 sur le facteur d'échelle du codage CPM. L'effet de correction est plus notable pour le codage basé-région que le codage non basé-région. La différence s'explique par le codage de blocs frontières, qui augmente la qualité dans ces régions. Mais quand le nombre de bloc frontière augmente, il devient intéressant d'utiliser le codage non basé-région que celui du basé-région.

Passant maintenant à la comparaison entre le codage basé-région avec celui de l'image entière en utilisant les courbes de la valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne (BW) pour une séquence de 20 trames. Chaque point représente une valeur de tolérance à partir de 1 avec un incrément de 2 jusqu'à 19 (10 points). Les figures (Fig.3.13), (Fig.3.14), (Fig.3.15) et (Fig.3.16) illustre les résultats trouvés sur les séquences *News*, *Children*, *Weather* et *Cyclamen* respectivement.

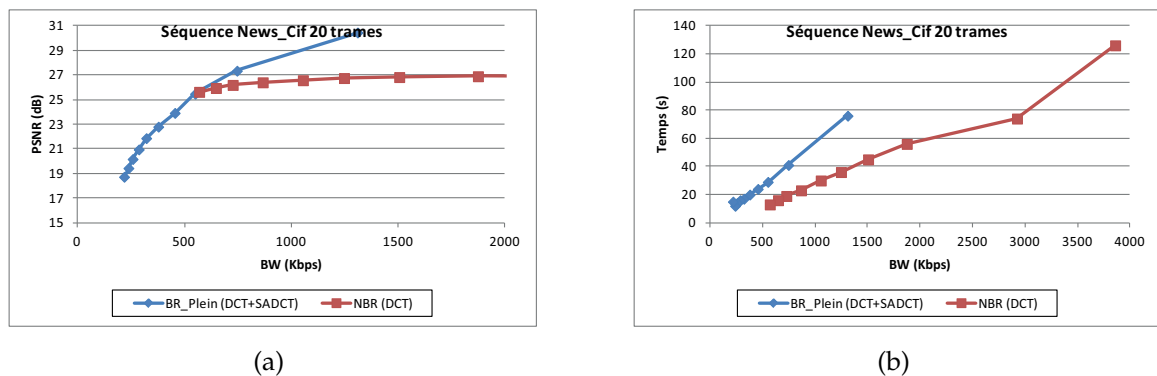
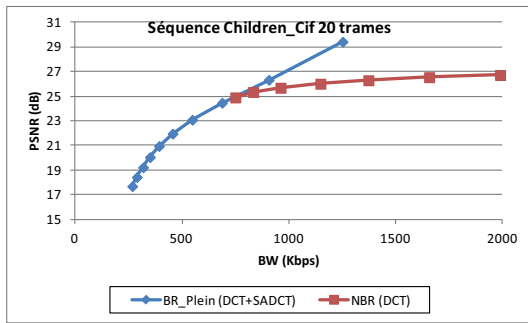


Figure 3.13 – Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence *News* de 20 trames. (a) La valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne (BW), (b) Temps de codage en fonction de la largeur de bande moyenne (BW).

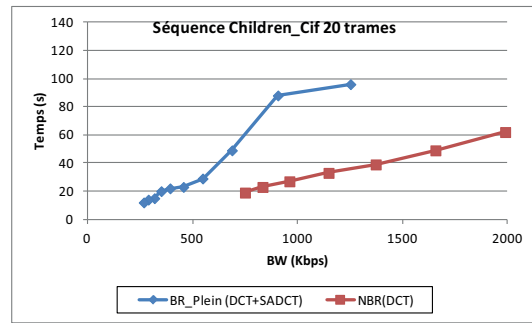
Dans la figure (Fig.3.13(a)), on remarque pour une largeur de bande supérieure à 500Kbps, les résultats du codage basé-région sont de meilleure qualité et peuvent atteindre jusqu'à 31dB alors pour le non basé-région, la qualité se stabilise à 27dB pour une largeur de bande moyenne qui dépasse 2000Kbps. Pour le temps de codage,(Fig.3.13(b)), le codage basé-région nécessite plus de temps ceci est dû au codage des blocs frontières qui vont se coder deux fois (deux blocs cibles).

Dans le même ordre d'idées, les figures ((Fig.3.14), (Fig.3.15) illustrent les résultats pour le même type d'expérience sur les séquences *Children* et *Weather* respectivement.

La figure (Fig.3.16) compare les résultats de la qualité moyenne en fonction de la largeur de bande moyenne pour la séquence *Cyclamen* de 20 trames. Le codage non basé-région peut atteindre une qualité supérieure par rapport au codage basé-région. Comme on a dit précédemment, ceci revient à la nature de la séquence *Cyclamen* qui présente beaucoup de

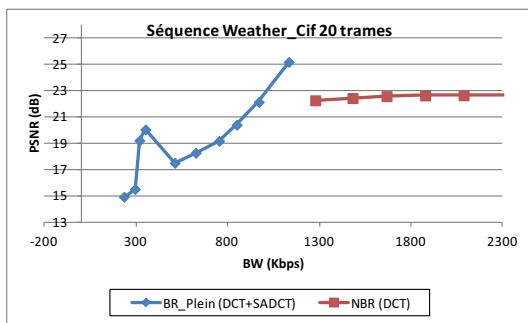


(a)

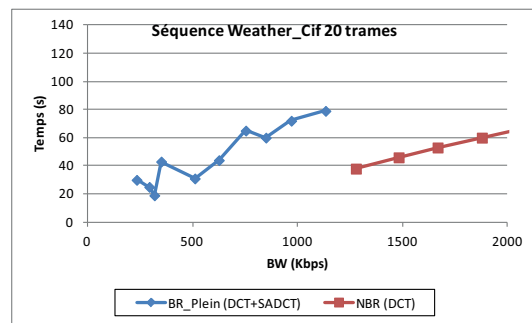


(b)

Figure 3.14 – Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence Children de 20 trames. (a) La valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne (BW), (b) Temps de codage en fonction de la largeur de bande moyenne (BW).



(a)



(b)

Figure 3.15 – Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région(NBR) dans le domaine transformé pour la séquence Weather de 20 trames. (a) La valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne (BW), (b) Temps de codage en fonction de la largeur de bande moyenne (BW).

blocs frontières.

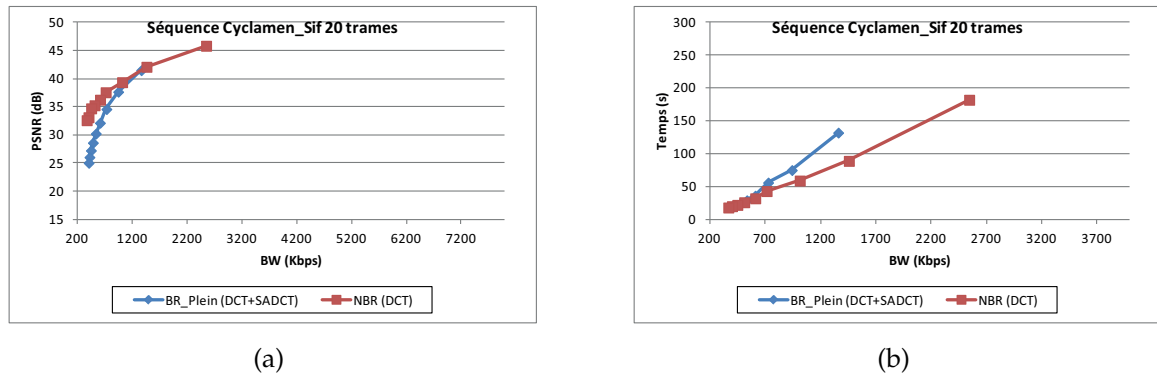


Figure 3.16 – Comparaison entre le codage fractal basé-région (Plein) et le codage fractal non basé-région (NBR) dans le domaine transformé pour la séquence *Cyclamen* de 20 trames. (a) La valeur moyenne de la qualité (PSNR) en fonction de la largeur de bande moyenne (BW), (b) Temps de codage en fonction de la largeur de bande moyenne (BW).

Après la confirmation des résultats trouvés dans [18], il nous reste une comparaison des résultats du codage fractal basé-région dans le domaine fréquentiel avec celui dans le domaine spatial. Pour ce faire, des expériences ont été faites sur les quatre séquences précédemment décrites qui comparent entre les résultats du codage fractal basé-région et non basé-région du domaine fréquentiel avec celui du domaine spatial. La tolérance utilisée pour ces expériences est de 6.5. Les figures (Fig.3.17, Fig.3.18, Fig.3.19 et Fig.3.20) illustrent les résultats sur les séquences *News*, *Children*, *Weather* et *Cyclamen* respectivement.

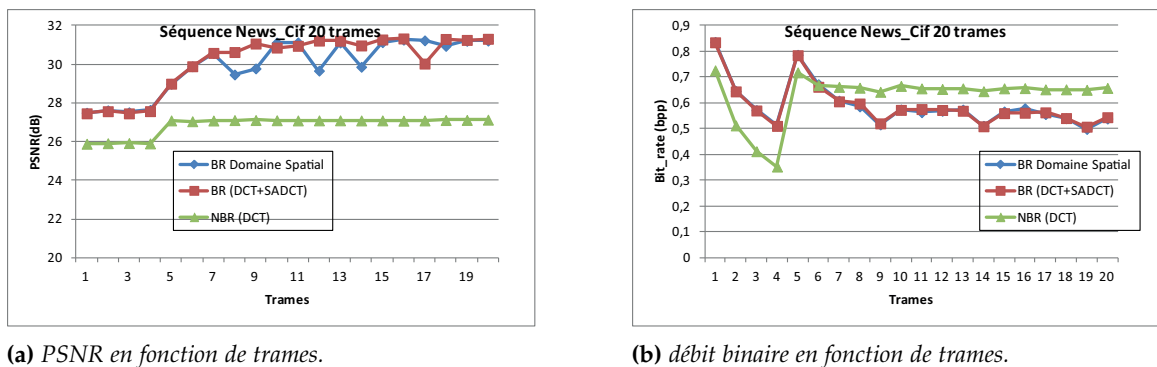
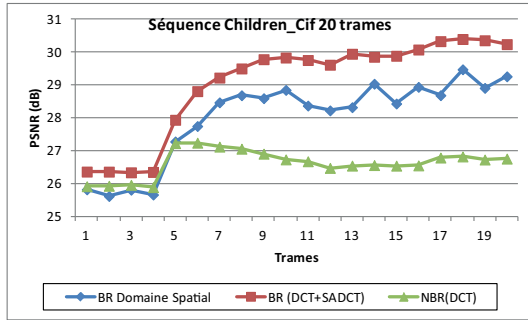
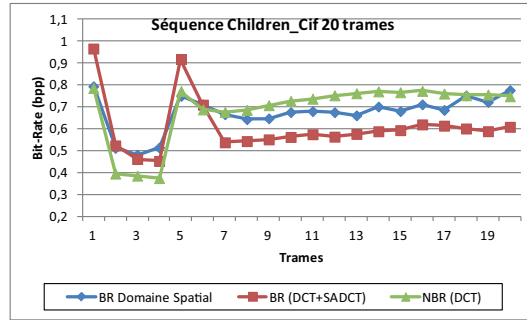


Figure 3.17 – Comparaison entre le codage fractal basé-région dans le domaine fréquentiel et le codage fractal basé-région dans le domaine spatial pour la séquence *News* de 20 trames.

Le codage fractal basé-région dans le domaine transformé présente une bonne qualité par rapport au codage fractal basé-région dans le domaine spatial. La séquence *Cyclamen* est une séquence vidéo spéciale qui présente beaucoup de bloc frontières, et pour cela le codage non basé-région présente une qualité meilleure en comparaison par le codage basé-région dans le domaine fréquentiel et celui dans le domaine spatial. Il revient plus facile à coder cette séquence par la méthode ordinaire que de la

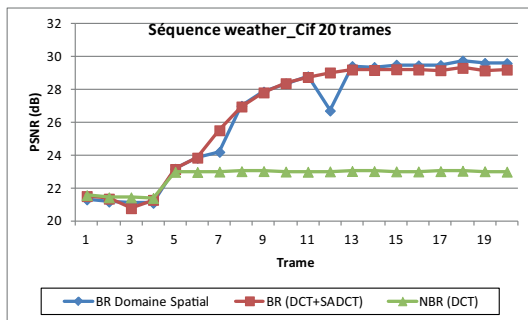


(a) PSNR en fonction de trames.

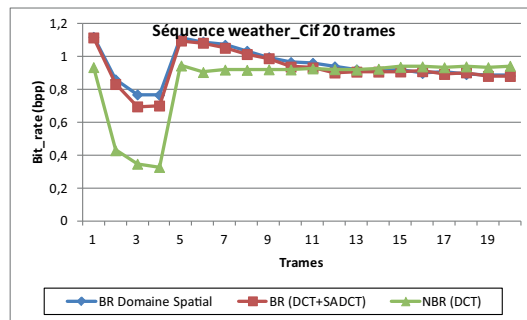


(b) débit binaire en fonction de trames.

Figure 3.18 – Comparaison entre le codage fractal basé-région dans le domaine fréquentiel et le codage fractal basé-région dans le domaine spatial pour la séquence Children de 20 trames.

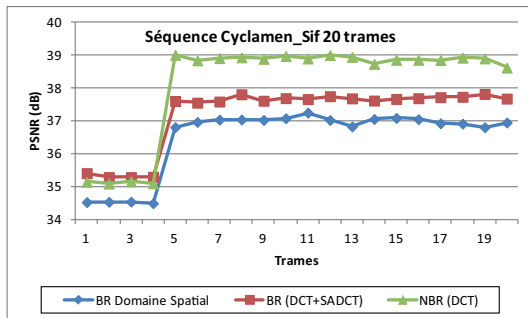


(a) PSNR en fonction de trames.

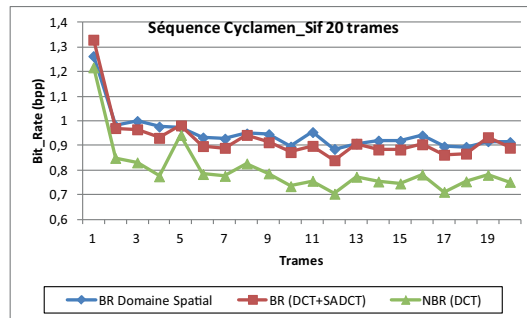


(b) débit binaire en fonction de trames.

Figure 3.19 – Comparaison entre le codage fractal basé-région dans le domaine fréquentiel et le codage fractal basé-région dans le domaine spatial pour la séquence Weather de 20 trames.



(a) PSNR en fonction de trames.



(b) débit binaire en fonction de trames.

Figure 3.20 – Comparaison entre le codage fractal basé-région dans le domaine fréquentiel et le codage fractal basé-région dans le domaine spatial pour la séquence Cyclamen de 20 trames.

coder segment par segment aux frontières.

Les figures (Fig.3.21(a)), (Fig.3.21(b)) et (Fig.3.21(c)) illustrent une comparaison subjective entre le codage fractal basé-région dans le domaine fréquentiel, le codage fractal

basé-région dans le domaine spatial et le codage fractal non basé-région dans le domaine fréquentiel pour la trame N° 5 de la séquence *Cyclamen*.

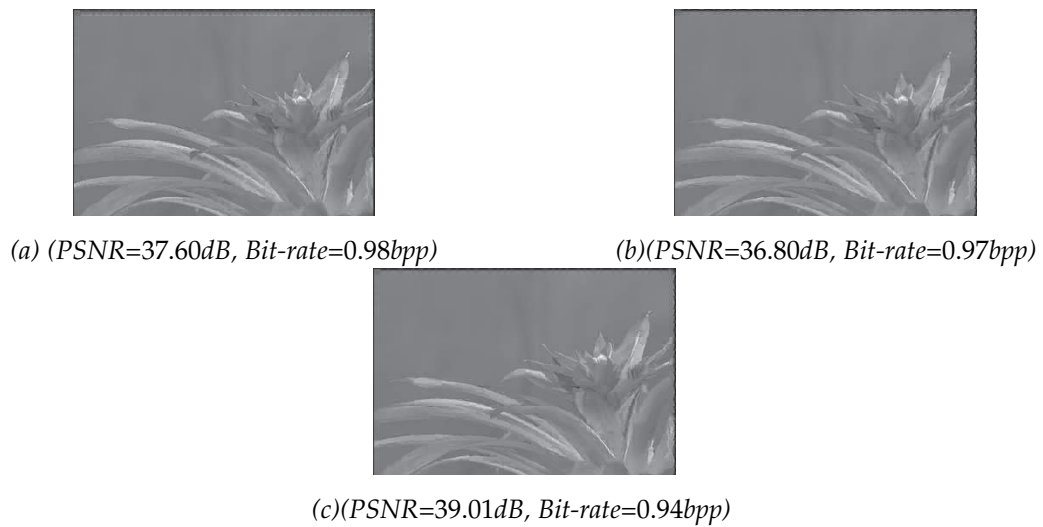


Figure 3.21 – Comparaison subjective entre (a) le codage fractal basé-région dans le domaine fréquentiel, (b) le codage fractal basé-région dans le domaine spatial et (c) le codage fractal non basé-région dans le domaine fréquentiel, pour la trame N° 5 de la séquence *Cyclamen*.

La figure (Fig.3.22) montre le premier-plan du décodage fractal basé-région et celui du non basé-région dans le domaine fréquentiel de la trame N° 5 de la séquence *Cyclamen*, preuve que le décodage a été effectué de façon indépendante pour chaque région avec le même alpha-plane disponible au décodeur.

On remarque une forte amélioration du décodage près des frontières pour la méthode du codage fractal basé-région.



Figure 3.22 – Trame N° 5 décodée de la séquence *Cyclamen*. (a) Premier-plan du codage fractal basé-région dans le domaine fréquentiel, (b) Premier-plan du codage fractal non basé-région dans le domaine fréquentiel.

3.6.1.1 Conclusion partielle

Les résultats présentés dans cette section démontrent une amélioration au niveau de la qualité et la largeur de bande comparativement aux résultats du codeur fractal dans le domaine spatial.

Puisque le codage de chaque objet est indépendant, il est maintenant possible de diminuer le taux de transmission des objets qui subissent très peu de mouvements ou de déformation.

Par contre, on a vu pour certains cas, que le codage fractal basé-région devenait plus nuisible que bénéfique. Par exemple, dans le cas où la séquence présente beaucoup de blocs frontières avec un mouvement rapide et il n'y a pas un fort contraste entre l'objet et le fond ce qui rend la tâche au NBR plus facile. Ceci dit, les deux segments d'un bloc frontière se ressemblent et par conséquent ils accepteraient d'être transformés (DCT) et codés ensemble. Donc, le codage basé-région se montrait inutile comparé à la méthode du codage non basé-région, la qualité devenait inférieure et la largeur de bande se voyait augmenter.

Dans ce qui suit on montre les résultats trouvés pour le codage de la stéréo avec la méthode proposée dans la section (Sec.3.5).

3.6.2 Codage fractal basé-région de séquence stéréo

Pour évaluer la performance du codec fractal des séquences stéréo proposées, on utilise deux séquences stéréo : *Anne* (352 x 288 pixels, 5 trames droites et gauches, 8.33 trames/s) et *Sergio* (256 x 256 pixels, 10 trames droites et gauches, 8.33 trames/s) et leurs alphas-planes. Les tailles maximales et minimales du partitionnement en quadtree pour le QDE/QDC sont 16 x 16 pixels et 4 x 4 pixels respectivement.

Les résultats de la moyenne du taux de compression pour *Anne* pour les vues droites et gauches sont illustrés dans le tableau (Tab.3.4) respectivement. Le temps de codage est de 6 min 18 s et du décodage est de 7 s.

	<i>Anne droite</i>			<i>Anne gauche</i>		
	Région0 [Objet Principal]	Région1 [Arrière-plan]	Plein	Région0 [Objet Principal]	Région1 [Arrière-plan]	plein
<i>PSNR (dB)</i>	32.05	38.41	35.17	30.67	39.50	34.40
<i>Compression Ratio</i>	9.34	17.26	13.46	15.50	16.94	16.41
<i>Bit per pixel</i>	0.42	0.20	0.62	0.25	0.24	0.49
<i>Bit Rate (kbps)</i>	705.07	765.07	1470.15	440.66	766.05	1206.71

Table 3.4 – Résultats du codage sur la séquence stéréo *Anne*

Les résultats de la moyenne du taux de compression pour *Sergio* pour les vues droites et gauches sont illustrés dans le tableau (Tab.3.5) respectivement. Le temps de codage est de 6 min 32 s et du décodage est de 8 s.

	<i>Sergio droite</i>			<i>Sergio gauche</i>		
	Région0 [Objet Principal]	Région1 [Arrière-plan]	plein	Région0 [Objet Principal]	Région1 [Arrière-plan]	Plein
<i>PSNR (dB)</i>	31.15	42.36	33.94	30.14	42.25	32.91
<i>Taux de Compression</i>	10.05	35.38	15.84	14.56	34.24	20.48
<i>Bit per pixel</i>	0.39	0.11	0.50	0.27	0.12	0.39
<i>Bit Rate (kbps)</i>	622.97	184.92	807.9	437.13	187.93	625.06

Table 3.5 – Résultats du codage sur la séquence stéréo *Sergio*

Une confirmation visuelle des revendications ci-dessus peut être trouvée dans les figures (Fig.3.23) et (Fig.3.24) où les résultats pour l'encodage *Anne* et *sergio* respectivement. On a choisi des paramètres de codage de façon à assurer une meilleure qualité du premier plan que pour l'arrière-plan. On a également sélectionné les paramètres de codage de manière à assurer une meilleure qualité de la vue de base (la vue de droite) que pour la vue d'amélioration (la vue de gauche).

La région0 décodée et la trame complète de la cinquième trame gauche et droite de *Anne* sont présentés dans la figure (Fig.3.23).

La région0 décodée et la trame complète de la cinquième trame gauche et droite de *Sergio* sont présentés dans la figure (Fig.3.24).

(a) *Premier-plan droit* (31.50 dB at 0.24 bpp)(b) *Premier-plan gauche* (30.20 dB at 0.18 bpp)(c) *Trame complète droite* (34.7 dB at 0.50 bpp)(d) *Trame complète gauche* (33.80 dB at 0.48 bpp)

Figure 3.23 – La cinquième trame décodée de la séquence stéréo Anne (a) *Premier-plan droit* (b) *Premier-plan gauche* (c) *trame complète droite* (d) *Trame complète gauche*.

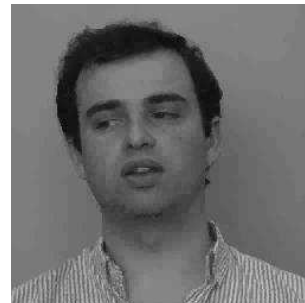
(a) *Premier-plan droit* (33.66 dB at 0.40 bpp)(b) *Premier-plan gauche* (30.77 dB at 0.30 bpp)(c) *Trame complète droite* (36.28 dB at 0.50 bpp)(d) *Trame complète gauche* (33.50 dB at 0.40 bpp)

Figure 3.24 – La cinquième trame décodée de la séquence stéréo Sergio (a) *Premier-plan droit* (b) *Premier-plan gauche* (c) *trame complète droite* (d) *Trame complète gauche*.

CONCLUSION

4

Basé sur la méthode de compression d'image fractale classique, un codec vidéo monoculaire et stéréoscopique est proposé dans le présent document. On a proposé une nouvelle vision, codec vidéo monoculaire et stéréo basé-région, qui emploie les fractales et le Shape-Adaptive DCT. Par conséquent, le décodeur peut décoder une région sans avoir référence à des informations sur d'autres régions de la vidéo. De toute évidence, ce qui permet des opérations intéressantes au décodeur, telles que les requêtes à base d'objets, la transmission d'un objet sélective ou même le remplacement de l'objet.

Le chapitre 2 de cette partie nous a introduits au codage de l'image. On a étudié la théorie des fractales de même que leur utilité pour coder les images fixes. On a introduit aussi les algorithmes du codage de la vidéo par fractales. On a expliqué les algorithmes CPM et NCIM et leur hybridation. À la fin de ce chapitre, on a détaillé le codage fractal basé-région de la vidéo dans le domaine spatial [18]. On a expliqué son principe et son fonctionnement.

Au chapitre 3, on a détaillé l'aspect théorique de la méthode proposée. On s'est d'abord familiarisé aux principes de base du codage par région. On a détaillé l'algorithme du SA-DCT et son application pour le codage fractal des séquences vidéo monoculaires puis pour les séquences stéréo. À la fin de ce chapitre, On a présenté les résultats expérimentaux. Les résultats présentés démontrent une amélioration au niveau de la qualité, comparativement à la méthode d'encodage fractal des séquences vidéo dans le domaine spatial. En effets, le codage basé-région dans le domaine transformé, a contribué à des gains d'environ 2dB pour la qualité (PSNR), avec une faible diminution de la largeur de bande dans certaines séquences comme la séquence *Children*. Par contre, pour certains cas, les codages basés-région dans le domaine transformé et celui dans le domaine spatial devenaient plus nuisibles que bénéfiques. Car dans ces cas, la qualité devenait inférieure et la largeur de bande augmente. Il serait intéressant de travailler avec les méthodes classiques de codage. En comparant avec le codec vidéo stéréo MPEG-4 [81], le codec fractal stéréo basé-région proposé a des performances (PSNR et débit) comparatives, et d'autres avantages, tels que des processus de codage et de décodage plus simples, véritables fonctionnalités indépendantes de codage / décodage basé-région, processus de décodage très rapide en raison des caractéristiques inhérentes du codage fractal, etc.

Les résultats expérimentaux montrent la justesse et l'efficacité, et un compromis entre un

taux de compression élevé et une bonne qualité (PSNR). En plus, ils sont comparables à ceux des standards MPEG et d'autres méthodes de référence dans le domaine.

Des efforts supplémentaires seront investis pour accélérer le processus de codage en cours en utilisant d'autres méthodes que celles de [91], [84].

II

Indexation Basée-Région d'Image Utilisant Shape-Adaptive DCT (SA-DCT)

INTRODUCTION



1.1 CONTEXTE

Les nouveaux supports informatiques tels que les tablettes et les smart-phones permettent aujourd'hui de manipuler des quantités très importantes d'informations. Parmi ces informations, les images prennent une part de plus en plus volumineuse. De nombreuses applications manipulent de manière quotidienne des images (multimédia, infographie, médicale, etc.). Il apparaît donc important de concevoir des systèmes d'indexations, capables de mener à bien la recherche d'informations au sein de très grandes collections d'images numériques. La recherche d'images par le contenu (CBIR) est le processus de recherche d'images selon leurs contenus visuels [72, 31]. L'extraction de caractéristiques est la base de CBIR, qui est le processus de transformation de l'image d'entrée en un ensemble de vecteurs de caractéristiques. Le vecteur de caractéristique peut être local ou global. Un descripteur global utilise des caractéristiques visuelles de l'image entière, on parle ainsi du CBIR, par contre un descripteur local utilise des caractéristiques visuelles des régions ou objets, on parle ainsi du RBIR. Pour obtenir un descripteur local on a besoin d'une segmentation a priori de l'image pour avoir des objets qui ont un sens (éléphant, cheval.....)[80]. Contrairement aux approches CBIR qui calculent un descripteur global pour les images, les approches RBIR extraits les caractéristiques des régions segmentées [77, 56, 54, 29]. Pendant la recherche, les systèmes RBIR fournissent le masque des régions segmentées de l'image requête ainsi que d'autres propriétés telles que la région recherchée, les caractéristiques des régions ainsi que le poids pour chaque région [48].

1.2 PROBLÉMATIQUE ET OBJECTIFS

L'indexation et la recherche des images sont un domaine très vaste, très riche et toujours d'actualité ; beaucoup de recherches sont en cours qui ont pour but d'améliorer les techniques d'indexation pour obtenir des résultats performants. Pour comparer entre les différentes méthodes existantes on considère les critères suivants :

- Les courbes précision-rappel.
- Le MAP (Mean Average Precisions).
- Le EER (Error Equal Rate).
- La rapidité de l'indexation ainsi que la vitesse de la recherche.
- La taille du descripteur qui doit être inférieure à la taille de l'image pour son stockage ou sa transmission (CDVS [21, 1, 2, 3]).

Le calcul du descripteur de l'image est le noyau d'un système d'indexation et de recherche d'image. Selon que ce descripteur est global ou local, on trouvera deux grandes familles : la recherche d'image par le contenu (CBIR) et la recherche d'image par région (RBIR). Les caractéristiques peuvent être extraites directement des images dans leurs formats compressés sans passer par le processus de décompression en utilisant par exemple la Transformée Cosinus Discrète (DCT) (Tel que nous l'avons présenté dans la partie I de cette thèse).

Dans la recherche d'image par le contenu classique (CBIR) qui utilise les blocs DCT [10], les blocs sont définis indépendamment du contenu de l'image. Les blocs contours contiennent des pixels de l'objet et d'autres de l'arrière-plan ou bien d'un autre objet qui se trouve à côté. Ainsi, l'indexation et la recherche indépendante des objets par cette méthode sont impossibles, en plus, la qualité de la recherche est très mauvaise. Aussi l'application de la DCT standard aux blocs frontières des objets entraîne une oscillation spectrale très importante.

Pour contourner ce problème, on propose une nouvelle approche de recherche d'image basée-région (RBIR) qui prend en considération une segmentation a priori de l'image en région [93]. Ceci est dans l'objectif d'améliorer la similarité à la granularité de la région indépendamment des autres régions (ou l'arrière-plan). Cette approche permet une recherche indépendante des objets individuels ce qui offre une nouvelle fonctionnalité. Donc, l'objectif du travail proposé est le développement d'une méthodologie optimisée d'indexation et de recherche d'image basée-région dans le domaine transformé en utilisant le Shape Adaptive DCT.

1.3 CONTRIBUTION

Notre travail propose une nouvelle approche dans le domaine de la recherche d'images par région (RBIR), utilisant la transformation cosinus discrète adaptée aux objets (SA-DCT). Dans ce système de recherche, une image a une segmentation a priori (alpha plane), tel qu'il a été défini dans MPEG-4 (Voir la partie I de cette thèse). Autrement dit, une image est représentée par des régions segmentées, et chaque région est associée à un vecteur de caractéristiques dérivés de la DCT et la SA-DCT. Chaque région de l'image requête, peut être sélectionnée comme l'objet principal. Pour les images qui n'ont pas un objet principal, l'utilisateur pourra toujours sélectionner l'image entière comme requête.

Les résultats obtenus montrent que l'approche proposée est capable d'identifier l'objet principal et de réduire l'influence de l'arrière-plan, ceci améliore la performance de la recherche d'images en comparaison avec d'autres systèmes conventionnels utilisant la DCT dans le domaine CBIR et même avec des systèmes dans le domaine RBIR.

1.4 STRUCTURE DE LA DISSERTATION

Dans le chapitre 2, on présentera le système d'indexation et de recherche d'image par le contenu basé sur la transformation DCT. On détaillera la manière pour calculer le descripteur des images par cette approche, l'histogramme des AC et l'histogramme des DC et leur combinaison. On parlera brièvement des différentes méthodes de mesure de similarité et d'évaluation de performance. Au chapitre 3, on proposera le système d'indexation et de recherche d'image par région basé sur la transformation SA-DCT. On détaillera le principe de l'approche proposée puis on présentera les résultats obtenus et on les analysera. On identifiera les améliorations et les inconvénients apportés par l'ajout de la SA-DCT. On conclura cette étude en soulignant les résultats importants obtenus et en dégageant les principales perspectives.

SYSTÈME D'INDEXATION ET DE RECHERCHE D'IMAGE PAR LE CONTENU UTILISANT LA DCT

2

2.1 INTRODUCTION

Pour les images compressées, les caractéristiques peuvent être extraites directement des images dans leurs formats compressés sans passer par le processus de décompression en utilisant par exemple la Transformation Cosinus Discrète (DCT). Récemment, plusieurs approches basées sur la DCT pour l'extraction de caractéristiques des images [35, 27, 34] ont été proposées. La DCT est de plus en plus utilisée car elle se caractérise par ses propriétés de dé-corrélation, la préservation des caractéristiques de l'image et la réduction de complexité. Il a été démontré dans certains cas que le traitement dans le domaine DCT donne des résultats nettement meilleurs que le traitement dans le domaine pixels [86]. Ce chapitre est organisé comme suit : la section (Sec.2.2) détaillera le principe de l'indexation d'image par le contenu. La section (Sec.2.3) présentera la méthode pour calculer le descripteur d'images dans le domaine de la DCT [9]. La section (Sec.2.4) parlera des différentes mesures utilisées pour calculer la similarité entre deux images. Les différentes méthodes utilisées pour évaluer la performance des systèmes sont vues dans la section (Sec.2.5). Une conclusion sera tirée dans la section (Sec.2.6).

2.2 PRINCIPE DE L'INDEXATION D'IMAGE PAR LE CONTENU

Les systèmes d'indexation et de recherche d'image par le contenu permettent de rechercher les images d'une base d'images en fonction de leurs caractéristiques visuelles. Un système de recherche d'images par le contenu typique est divisé en deux parties : l'extraction de caractéristiques qui se fait en mode différé (off-line) et la recherche d'images qui se fait en temps réel (on-line). La figure (Fig.2.1) présente l'architecture générale d'un système de recherche d'images par le contenu (CBIR). Dans l'étape off-line, le système extrait automatiquement les attributs visuels, encore appelées caractéristiques de bas-niveau (couleur, forme, texture, et l'information spatiale) de chaque image de la base de données en fonction de ses valeurs des pixels spatiales ou bien des coefficients de sa transformée. Ensuite, ces attributs ou bien descripteurs seront stockés dans une base de données pour faire la recherche par la suite. Les données de caractéristiques (aussi connu comme la signature d'image) pour chacun des attributs visuels de chaque image sont beaucoup plus petits par rapport à l'image elle-même. Ceci indique que la base de données des vecteurs de caractéristiques contient une version compacte de la base des données d'image. Un avantage d'une signature sur les valeurs d'origine de pixel est la compression importante de représentation d'image. Cependant, une raison plus importante pour l'utilisation de la signature est de gagner une meilleure corrélation entre la représentation de l'image et de la sémantique visuelle.

Dans la recherche d'image on-line, l'utilisateur peut soumettre un exemple de requête pour le système pour la recherche d'images souhaitées. Le système extrait de cette requête un vecteur de caractéristiques. Les distances entre les vecteurs de caractéristiques de la requête et ceux de la base de données sont ensuite calculées et classées. La recherche est réalisée par l'application d'un schéma d'indexation qui fournit un moyen efficace de recherche dans la base de données d'image. Enfin, le système classe les résultats de la recherche, puis renvoie les résultats qui sont les plus similaires à la requête. Si l'utilisateur n'est pas satisfait avec les résultats de la recherche, il peut fournir un feedback pour le système de recherche, qui contient un mécanisme pour apprendre les besoins de l'utilisateur.

Pour pouvoir comparer des caractéristiques visuelles, il faut que la représentation des caractéristiques visuelles obtenue soit :

- discriminante : elle doit permettre de bien différencier les images différentes.
- invariante : deux régions de tailles différentes ou prises avec des luminosités différentes doivent avoir des représentations très proches.
- compacte : les espaces de grandes dimensions sont sensibles à la malédiction de la dimension et posent des problèmes de stockage des données. Il est donc intéressant de réduire au maximum la dimension des signatures.

Une représentation pratique est l'histogramme. Cet outil est plutôt insensible aux changements d'orientation, de taille et de position des régions, mais il ne capture pas les relations spatiales entre les régions, et par conséquent a un pouvoir discriminant limité. Un histo-

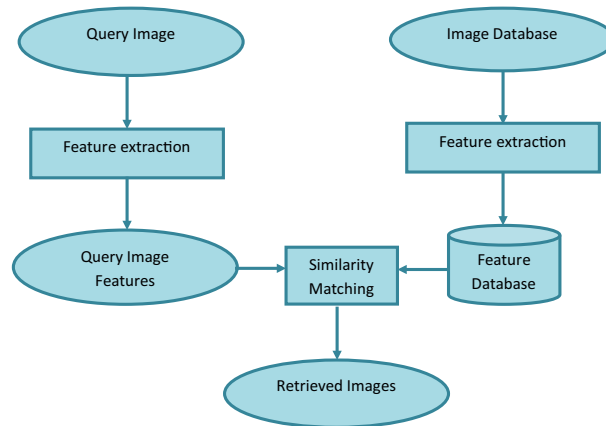


Figure 2.1 – Architecture générale d'un système d'indexation et de recherche d'image par le contenu (CBIR).

gramme peut être vu comme le nombre d'apparitions d'un élément dans un ensemble. Les caractéristiques peuvent être extraites directement des images compressées sans passer par le processus de décompression. Dans la section (Sec.2.3), une méthode d'indexation et de recherche d'image par le contenu dans le domaine transformé utilisant la DCT sera bien détaillée.

2.3 DESCRIPTEURS D'IMAGES DANS LE DOMAINE DCT

Supposons que I est l'intensité de l'image qu'on veut transformer et $\mathbf{x} = (x, y)$ sont les coordonnées spatiales pour les pixels de cette image. Supposons que $\{b_1, \dots, b_N\}$ est l'ensemble des blocs qui ne se chevauchent pas, c.à.d, la collection de coordonnées de pixels, qui divisent l'image. Pour permettre une notation compacte, supposant que :

$$I_{b_i} = \{I(x) : x \in b_i\} \quad (2.1)$$

Chaque bloc b_i est transformé dans le domaine fréquentiel $\widehat{I}_{b_i} = DCT(I_{b_i})$ par la transformation DCT. Cette transformation est initialement définie dans la forme 1-D, et peut-être utilisé pour construire une transformation 2-D séparable. Cette transformation est utile pour le codage source, particulièrement le codage de l'image et de la vidéo. $U = (u, v)$ sont les coordonnées fréquentielles des coefficients DCT et $\widehat{I}(u)$ est la transformée DCT pour une image $N \times N$ représentée par les pixels $I(x)$ tel que $x, y = 1, \dots, N$.

2.3.1 Histogramme des AC

Dans cette étude, on considère des blocs DCT 4×4 . Cette méthode prend en compte neuf coefficients AC parmi les quinze coefficients pour chaque bloc et utilise les informations statistiques pour construire les modèles AC (AC-pattern) (Fig.2.2). Ces 9 coefficients sont regroupés en 3 groupes : Horizontal(groupe H), Vertical (groupe V) et Diagonal (groupe D). Pour chaque groupe, on calcule premièrement la somme des coefficients. Ensuite, la différence des carrées entre chaque coefficient et la somme de ce groupe. Finalement, cette différence carrée des trois groupes est utilisée pour construire le modèle des AC (AC-Pattern). Cette sélection est retenue parce qu'elle est capable de représenter la structure interne du contenu du bloc [9]. En comparaison avec la méthode de [86] qui utilise 15 AC, cette sélection réduit la complexité du vecteur de caractéristique.

Pour construire l'histogramme des AC-patterns pour une image il suffit de calculer le nombre d'apparence de cet AC-pattern dans cette image, et puis de construire l'histogramme des AC-patterns dénommé H_{AC} . Par contre, il y a un AC-Pattern spécial là où tous les coefficients sont nuls (Fig.2.3), cet AC-Pattern correspond généralement aux blocs uniformes de l'image [9]. En considération du temps et de l'efficacité, on sélectionne seulement les AC-Patterns qui représentent les hautes fréquences pour construire l'histogramme [86].

2.3.2 Histogramme des DC

Contrairement au AC-pattern qui décrit les caractéristiques locales à l'intérieur de chaque bloc (Intra-bloc), le DC-pattern décrit les caractéristiques globales en utilisant le gradient entre chaque bloc et ses voisins (Inter-bloc). DC-DirecVec [86] est défini et utilisé comme caractéristique pour le DC-pattern. Pour une valeur DC, la différence entre cette valeur et ses huit autres DC voisins est calculée et arrangée pour produire le DC-pattern.

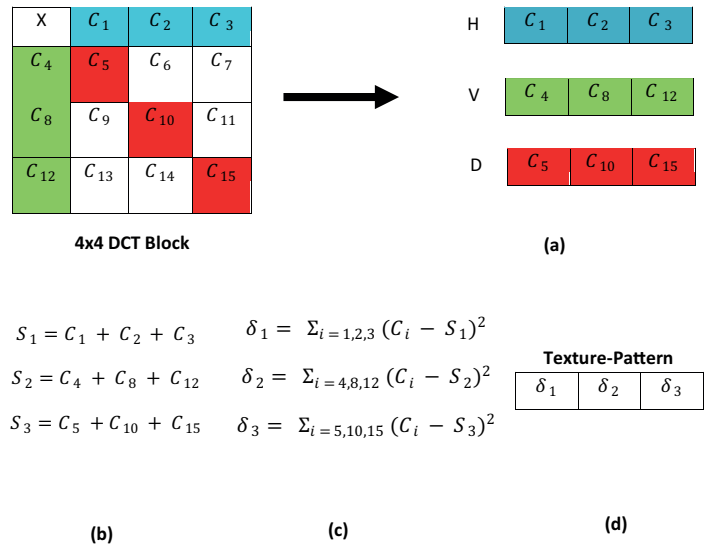


Figure 2.2 – Processus de construction du AC-pattern [9]

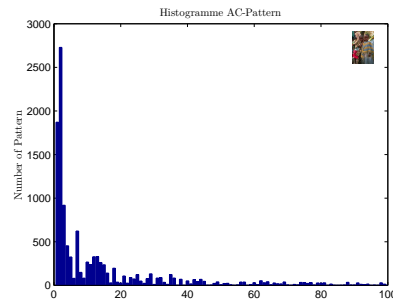


Figure 2.3 – Histogramme des 100 AC-patterns, hautes fréquences des occurrences de l'image African (CBIR)

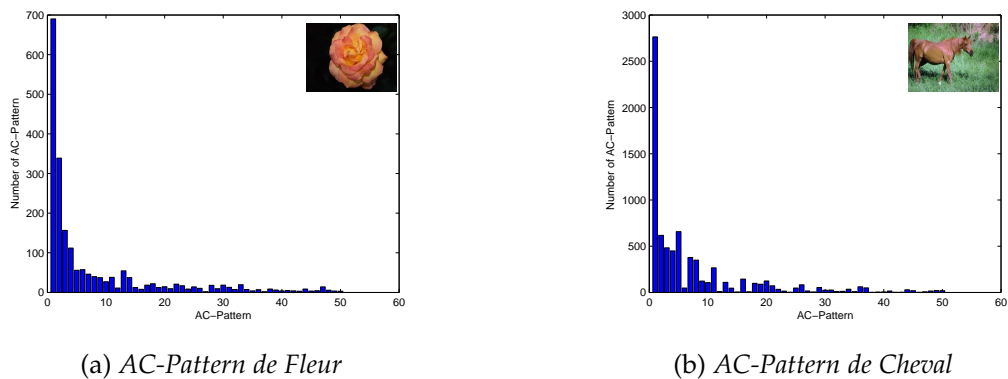


Figure 2.4 – Histogramme des 50 AC-patterns, hautes fréquences des occurrences de l'image (CBIR) (Content-Based Image Retrieval).

Plus précisément, DC-pattern est défini comme un ensemble de directions ayant les plus grandes différences entre la valeur du bloc courant et les valeurs des DC des blocs voisins. Huit différences entre le coefficient DC du bloc courant et les valeurs des DC des blocs voisins. La neuvième est la différence entre la valeur DC du bloc courant et la moyenne

de toutes les valeurs DC des neuf voisins (lui-même inclu). Les valeurs absolues de ces différences sont rangées par ordre décroissant et les γ premières directions avec les plus grandes différences forment le DC-pattern. γ est un paramètre qui peut être réglé pour améliorer la précision de la recherche. La figure (Fig.2.5) illustre le processus de construction du DC-pattern.

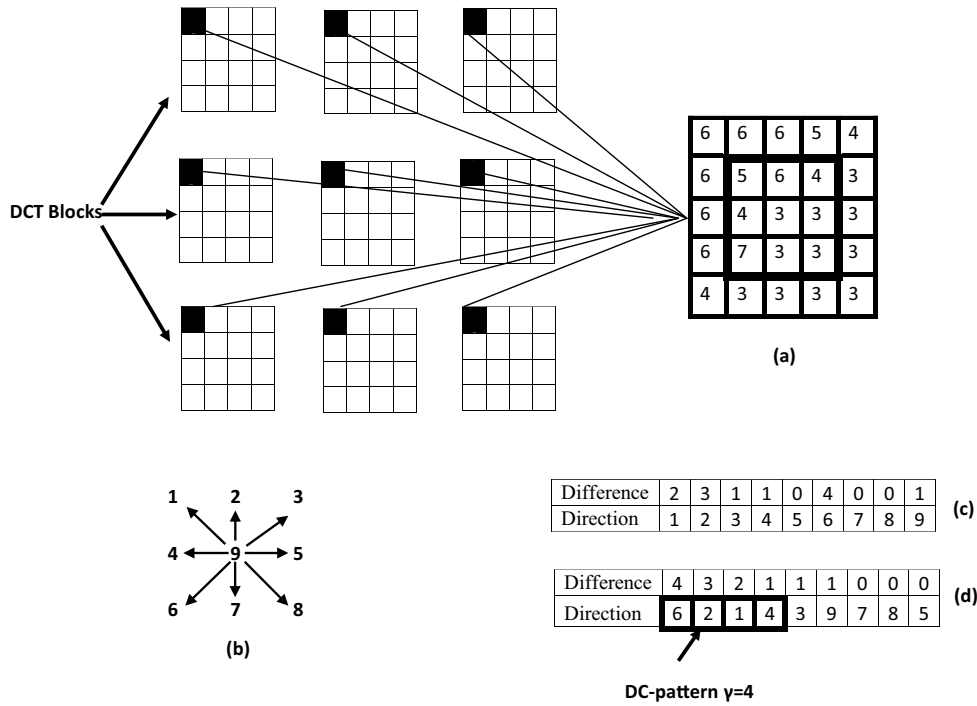


Figure 2.5 – Processus de construction du DC-pattern [86]

Comme on a dit pour l’histogramme des AC-patterns H_{AC} , on sélectionne les DC-patterns dominants pour construire l’histogramme des DC-patterns H_{DC} .

2.3.3 Descripteur de caractéristique

Pour chaque bloc, le AC-pattern est formé par 9 coefficients et le DC-pattern est construit par le coefficient DC du bloc lui-même et la différence entre cette valeur et ses 8 autres DC voisins. L’histogramme concaténé des deux histogrammes, H_{AC} et H_{DC} , représente l’index de l’image et il est utilisé pour faire la recherche (Fig.2.6). Dans ce contexte, le descripteur est défini comme suit :

$$H = [(1 - \alpha) * H_{AC}, \alpha * H_{DC}] \tag{2.2}$$

Où α représente le poids qui contrôle l’impact des histogrammes AC-pattern et DC-pattern. Ce paramètre peut être réglé pour améliorer la précision de la recherche.

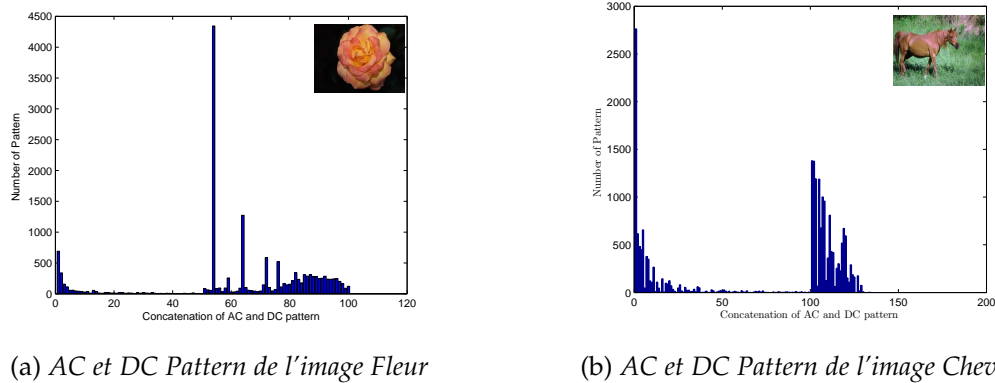


Figure 2.6 – Histogramme combiné des premières occurrences de hautes fréquences du AC-patterns et DC-patterns pour le CBIR.

2.4 MESURE DE SIMILARITÉ

Le vecteur de caractéristique peut être vu comme un vecteur normalisé. Ceci permet une comparaison des informations statistiques des images dans un chemin quantitatif par le calcul de la distance entre les histogrammes en utilisant une norme acceptable. Plusieurs distances ont été utilisées pour définir la similarité entre deux histogrammes [36]. Dans cette section, on détaillera les mesures de similarités entre deux histogrammes H_Q et H_D . En général, les mesures de similarité entre deux histogrammes peuvent être divisées en deux groupes : la mesure de similarité *coefficient à coefficient* et la mesure de similarité *croisée entre coefficients*. Les mesures de similarité *coefficient à coefficient* comparent les coefficients des histogrammes du même indice, c.à.d, ils comparent $H_Q(i)$ et $H_D(i)$ pour i , mais pas $H_Q(i)$ et $H_D(j)$ pour $i \neq j$. Par contre les mesures de similarité *croisées entre coefficients* comparent les coefficients qui ne correspondent pas.

2.4.1 Mesure de similarité coefficient à coefficient

Cette catégorie de mesure compare les coefficients qui se correspondent des histogrammes H_Q et H_D . La similarité entre deux histogrammes est la combinaison de ces comparaisons coefficient à coefficient.

2.4.1.1 Distance de Minkowski

La distance de Minkowski est défini comme suit :

$$d_{L_p}(H_Q, H_D) = \left(\sum_{i=1}^N |H_Q(i) - H_D(i)|^p \right)^{\frac{1}{p}} \quad (2.3)$$

Où $p=1,2$ ou ∞ , cette distance est toujours référée par la distance Manhattan, la distance euclidienne ou bien la distance Chebyshev respectivement. Ces trois distances sont les plus utilisées dans la recherche d'images.

2.4.1.2 Intersection d'histogramme

L'intersection d'histogramme est défini comme :

$$d_{\cap}(H_Q, H_D) = \sum_{i=1}^N \min(H_Q(i), H_D(i)) \quad (2.4)$$

Notons que cette distance est moins complexe car elle est constituée seulement de minimums et d'additions.

2.4.1.3 Distance Chi-square

La distance Chi-Square (χ^2) est utilisée pour comparer deux ensembles de données et pour déterminer s'ils sont tirés de la même fonction de distribution.

$$d_{\chi^2}(H_Q, H_D) = \sum_{i=1}^N \frac{(H_Q(i) - H_D(i))^2}{H_Q(i) + H_D(i)} \quad (2.5)$$

Où i représente la composante du descripteur et H_Q, H_D représentent les différents descripteurs, N indique la dimension du descripteur. Notons que dans notre système, la similarité entre l'image requête et une image dans la base de données est estimée par la distance Chi-Squared (éq.2.5).

2.4.2 Mesure de similarité croisée entre coefficients

Les mesures vues précédemment permettent la comparaison par la considération de la différence entre les coefficients des histogrammes de même rang. Et comme l'histogramme traditionnel a des discontinuités dans les coefficients des frontières, un petit décalage dans les valeurs de caractéristiques se produira. Alors, pour comparer ces histogrammes, d'autres distances existent.

2.4.2.1 Distance de forme quadratique

La distance de forme quadratique (QF) est défini comme suit :

$$d_{QF}(H_Q, H_D) = \sqrt{(H_Q - H_D).A.(H_Q - H_D)^T} \quad (2.6)$$

Où $A = [a_{ij}]$ est la matrice de similarité, et a_{ij} la distance entre les coefficients $H_Q(i)$ et $H_D(j)$. Les sélections de A les plus utilisées sont :

$$a_{ij} = 1 - \frac{d_{ij}}{d_{max}} \\ \text{ou bien } a_{ij} = \exp\left(-\beta \left(\frac{d_{ij}}{d_{max}}\right)^2\right)$$

Où β est une constante positive, et d_{ij} est la distance euclidienne entre les coefficients de $H_Q(i)$ et $H_D(j)$, d_{max} est le maximum de d_{ij} .

2.5 ÉVALUATION DE PERFORMANCES

Pour évaluer et comparer la performance des différents systèmes d'indexation et de recherche d'images, une évaluation de leurs performances est nécessaire. Ceci permet aux chercheurs de bien comprendre les limitations de leurs algorithmes et de comparer leurs résultats avec d'autres algorithmes objectifs. Dans cette section, on parle de certaines mesures d'évaluation de performance des systèmes d'indexation et de recherche d'images.

2.5.1 Précision et rappel

La mesure d'évaluation de performance la plus couramment utilisées pour le CBIR et le RBIR est la courbe précision-rappel. La précision mesure la capacité du système à fournir un maximum d'images pertinentes sur l'ensemble des images qu'il fournit et elle est définie comme le rapport entre le nombre des images pertinentes sur les r premières images et le nombre des images (r). Le rappel correspond à la faculté d'un système de retrouver les images pertinentes de la base de données par rapport à une requête et il est défini comme le rapport entre le nombre d'images pertinentes parmi les r images et le nombre total d'images pertinentes dans la base. Le système qui donne une meilleure précision pour le même rappel est le système le plus performant.

$$Precision = \frac{r}{r + s} \quad (2.7)$$

$$Rappel = \frac{r}{r + t} \quad (2.8)$$

Où r est le nombre d'images pertinentes récupérées, s est le nombre d'images non pertinentes récupérées, t est le nombre d'images pertinentes non récupérées de la base de données.

2.5.2 MAP (Mean Average Precision)

La précision moyenne (Mean Average Precision, MAP) est le moyen pour transformer le graphe précision-rappel en une seule valeur. Le MAP, pour toutes les requêtes, est défini par :

$$MAP = \frac{1}{Q} \sum_{q \in Q} AP(q) \quad (2.9)$$

Où Q est l'ensemble des requêtes et, $AP(q)$ est défini comme :

$$AP(q) = \frac{1}{N_R} \sum_{n=1}^{N_R} P(R_n) \quad (2.10)$$

Où R_n est le rappel après que la n^{ime} image pertinente est sélectionnée. $P(R_n)$ est la précision quand le rappel est R_n . MAP contient, en plus de la précision et du rappel, la position de l'image pertinente.

2.5.3 Taux moyen de récupération (Average Retrieval Rate, ARR)

Le taux moyen de récupération est utilisé dans la littérature pour les systèmes de recherche des images textures. Le taux de récupération (Retrieval Rate, RR) pour une requête est défini comme le pourcentage du nombre d'images pertinentes récupérées sur le nombre total d'images pertinentes dans la base de données, observées dans les K premières images extraites :

$$RR = \text{rappel} = \frac{r}{r+t}$$

ARR se définit comme la valeur moyenne de l'ensemble des taux de récupération des K premières images trouvées à chaque requête.

2.5.4 Taux d'erreur égale (Equal Error Rate, EER)

Le taux d'erreur égal est souvent utilisé pour évaluer la performance des systèmes de reconnaissance de visage. Après que la recherche est effectuée, il faut mesurer la similarité entre les images. Deux images sont similaires si la distance entre leurs vecteurs de caractéristiques est inférieure à un seuil. Deux taux d'erreur peuvent être définis, le taux de faux rejet (False Rejected Rate, FRR) qui résulte du fait qu'une image peut être rejetée à tort de sa classe, et le taux de fausse acceptation (False Accept Rate, FAR) qui résulte du fait qu'une image peut être acceptée à tort dans une classe. Ces deux taux varient en fonction du seuil. Lorsque FRR et FAR prennent des valeurs égales, un EER est obtenu. La performance d'un système est bonne si cette valeur est faible.

2.6 CONCLUSION

Dans ce chapitre, une méthode d'indexation d'images par le contenu dans le domaine transformé est introduite.

Tout d'abord, on a introduit le concept du CBIR. On a parlé de l'architecture générale d'un système d'indexation et de recherche d'images par le contenu. On a détaillé dans la section qui suit une approche dans le domaine de la DCT. C'est une approche basée sur une méthode dans laquelle l'histogramme des AC et celui du DC sont utilisés comme descripteurs de caractéristique. Après la construction des descripteurs d'images, la mesure de similarité doit être considérée ; ainsi, on a présenté les mesures de similarité les plus utilisés. La distance chi-square est choisie pour notre approche car elle est moins complexe.

Enfin, pour évaluer la performance des systèmes d'indexation et de recherche d'images et pour les comparer avec d'autres systèmes existants, des mesures d'évaluation ont été présentées. En s'appuyant sur la précision et le rappel avec ARR et EER car ils sont largement utilisés dans ce domaine.

Le chapitre suivant présentera la méthode proposée d'indexation et de recherche d'image basée-région utilisant la SA-DCT.

SYSTÈME D'INDEXATION ET DE RECHERCHE D'IMAGE BASÉ-RÉGION UTILISANT LA SA-DCT

3.1 INTRODUCTION

On propose d'appliquer la DCT adaptée aux objets (Shape-Adaptive DCT : SA-DCT)[71] pour le RBIR. Dans le CBIR qui utilise les blocs DCT [10] (Sec.2.2), les blocs contours contiennent des pixels de l'objet et de l'arrière-plan ou bien d'un autre objet qui se trouve à côté. Pour contourner ce problème, nous proposons l'application du SA-DCT qui prend en compte une segmentation a priori de l'image en région [24]. La recherche des régions est limitée à l'objet, c.à.d une région de l'image peut seulement correspondre à une région d'une autre image.

Ce chapitre est organisé comme suit, la première section (Sec.3.2) présentera quelques travaux connexes à l'indexation et la recherche d'image basée-région. Notre système RBIR sera présenté dans la section (Sec.3.3). Dans la section (Sec.3.4), on présentera les résultats expérimentaux et on les analysera. Une conclusion sera tirée dans la section (Sec.3.5).

3.2 INDEXATION D'IMAGE BASÉE-RÉGION

Plusieurs méthodes ont été développées. On ne peut pas citer toutes les méthodes existantes, par contre, on essaie de résumer certains travaux qui sont relatifs à notre approche.

Il existe des systèmes CBIR dans le domaine compressé utilisant un descripteur global, tel que l'histogramme des coefficients DCT [86, 87](Sec.2.3), pour représenter les images. Ils ont utilisé l'histogramme pour extraire les informations de la distribution des caractéristiques, sans avoir des informations sur la location de ses caractéristiques. Ces caractéristiques globales sont extraites de l'image entière et souvent ne peuvent pas capturer les détails locaux existant dans les images naturelles. Par conséquent, des images non similaires qui varient dans les détails locaux, peuvent avoir un vecteur de caractéristiques similaires. Cette anomalie crée des difficultés pour que la représentation globale ait une relation directe avec la sémantique de l'image.

Les systèmes de recherche basés-région tentent de pallier les lacunes des systèmes de recherche par le contenu. Un système de recherche basé-région applique la segmentation aux images pour la décomposer en région, qui correspond à des objets si la décomposition est idéal [77]. La représentation en objet est destinée à être proche de la perception du système visuel humain (SVH). Étant donné que le système de recherche identifie les objets qui se trouvent dans l'image, il sera plus facile pour ce système de reconnaître des objets semblables à des emplacements différents et avec des orientations et de tailles différentes. Les systèmes de recherche basés-région comprennent le système Netra [58], et le système Blobworld [25]. Les systèmes Netra et Blobworld comparent les images région par région. La raison est de transférer une partie de la tâche de comparaison aux utilisateurs. Pour chercher une image, un utilisateur dispose des régions segmentées de l'image et il est demandé de sélectionner les régions qui contribuent à la recherche et aussi les attributs, par exemple, la couleur et la texture, des régions utilisées pour évaluer la similarité.

Récemment, Natsev et al. considèrent le modèle WALRUS [8], qui est un modèle robuste d'extension et de la représentation d'objets dans une image. Chaque image est d'abord décomposée en région. La mesure de similarité entre les deux images est alors définie comme la proportion de la surface des deux images correspondant à la recherche des régions.

Pour mesurer la similarité entre les images, Li et Wang et al [77], ont proposés l'algorithme Region Integrated Matching (IRM), qui permet de faire correspondre une région d'une image à plusieurs régions d'une autre image. Autrement dit, la correspondance des régions entre deux images est une relation plusieurs à plusieurs. Par conséquent, la similarité entre les deux images est définie comme la somme pondérée des distances dans l'espace de caractéristiques, entre toutes les régions des différentes images. Par rapport

aux systèmes de recherche basées sur différentes régions, comme Blobworld, l'approche IRM diminue l'impact de la segmentation imprécise. IRM intègre les propriétés de toutes les régions segmentées afin que les informations sur une image peuvent être pleinement utilisées. Pour augmenter la robustesse contre les erreurs de segmentation, IRM compare une région avec plusieurs régions dans une autre image. Chaque région est affectée à un poids d'importance, ce qui correspond à l'importance de la région. Il y a plusieurs façons pour attribuer le poids à une région. Certains supposent que toutes les régions ont la même importance. Dans IRM, les objets importants dans une image ont tendance à occuper des zones plus larges, appelés un schéma basé sur le pourcentage de la surface.

Une autre méthode nommée ARM (Adaptive Region Matching), qui consiste à traiter le problème de mesure de similarité dans la recherche d'image basée-région (RBIR) a été proposé par Yang, X. et Cai, L. [82]. Une nouvelle mesure de similarité visuelle appelé recherche adaptée aux régions (ARM) a été développée. Pour diminuer l'influence négative de régions en interférences et la perte importante d'informations simultanément, un indice d'importance de région (RII, Region Importance Indice) est calculé pour trouver la région de signification sémantique (SMR). De plus, le ARM exécute automatiquement la recherche SMR-à-image ou bien la recherche image-à-image selon qu'il ait une région dominante ou non. Des expériences sur les bases de données Corel-1000 et Caltech-256 démontrent que le ARM proposé est plus flexible et plus efficace que les mesures de similarités visuelles existantes qui ont été à l'origine développées pour RBIR.

Par contre, le modèle SRBIR [60], identifie une région d'importance dans une image en utilisant un mécanisme basé sur l'attention visuelle et représente cette région en utilisant les descripteurs couleurs et les descripteurs basés sur la transformé en curvelet. Ces caractéristiques sont extraites de la région significative de l'image requête et des images de la base de données. Les similarités entre la région de l'image requête et les régions des images de la base de données sont classées selon une mesure calculée à partir des vecteurs de caractéristiques. Le modèle proposé n'a pas besoin d'une compréhension sémantique complète du contenu de l'image, il utilise les paramètres visuels tels que la proximité, la taille, le contraste des couleurs et la proximité aux frontières d'image pour localiser la région significative dans l'image et utilise la transformé en curvelet en combinaison avec les descripteurs couleurs pour représenter cette région.

Toutefois, Zhong et Defee [88] ont proposé une approche pour unifier les informations statistiques et structurelles pour la recherche d'images. Ils incorporent une description des informations structurelles en patterns par la considération de la décomposition des patterns en *régions carrées*. La similarité entre les images est donc déterminée par la comparaison des signatures de ces sous images. Les inconvénients de ce système sont la complexité de calcul et l'augmentation de la taille de l'espace de recherche due à la

génération exhaustive des sous images.

Pour permettre d'extraire les caractéristiques des régions arbitrairement formées pour les systèmes RBIR, Liu et al. [57, 55] utilisent une approche itérative basée sur la théorie des projections successives pour les ensembles convexes (Projections Onto Convex Sets, POCS). L'algorithme POCS est une technique itérative et le processus itératif sera terminé quand les pixels en dehors des frontières convergent. L'algorithme peut être déterminé par les étapes suivantes :

1. Extrapoler une partie indéterminé (Fig. 3.1) dans \mathcal{U} de \mathcal{S} (pour obtenir un signal rectangulaire) ;
2. Calculer la DCT 2-D pour le rectangle ;
3. Exit si le résultat est satisfaisant ;
4. Remettre à zero quelques coefficients de la DCT (pour limiter la largeur de bande du signal) ;
5. Calculer la DCT 2-D inverse pour le signal limité en largeur de bande ;
6. Dans \mathcal{S} , remplacer le résultat trouvé avec le signal original pour assurer un signal non-modifiable à l'intérieur de \mathcal{S} ;
7. retourner à 2).

La théorie des POCS converge seulement si la sélection des coefficients DCT retenus est fixée avant les itérations. Cela veut dire que la sélection des coefficients retenus est hautement dépendante de l'extrapolation initiale ; les composantes de haute fréquence doivent être produites par une extrapolation souhaitable, et ceci met les résultats dans une zone sous optimal des coefficients qui doivent être retenues. Dans [57], ils utilisent la transformation des ondelettes (DWT) au lieu de la DCT. Dans la compression d'image basée-objet, Stasinski et Konrad [73] ont montré la performance de la DCT adaptée aux objets (Shape-adaptive DCT, (SA-DCT)) en comparaison avec POCS.

3.3 PRINCIPE DE L'APPROCHE PROPOSÉE

Dans notre approche, pour résoudre le problème de la méthode [88], on cherche des techniques qui divisent les images en région sémantiques et que la représentation de ces régions doit capturer les détails locaux. Ces techniques sont connues sous le terme de la recherche d'images par région (Region-Based Image Retrieval, RBIR). Toutefois, toutes les techniques RBIR fonctionnent de la même façon, diviser les images en région et les représenter en matière de régions de caractéristiques. Dans la méthode classique basée sur le contenu (CBIR) (Section.2), les blocs sont définis indépendamment du contenu des images (Fig.3.1). Alors, une recherche indépendante des objets par cette méthode est impossible.

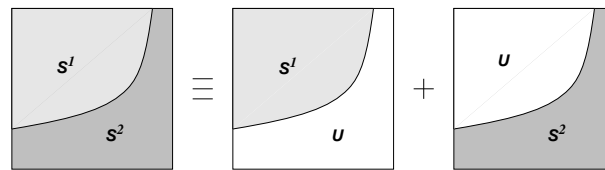


Figure 3.1 – Bloc avec deux segments S^1 et S^2 .

Toutefois, la qualité de la recherche souffre des pixels de différentes intensités ; et l'application de la DCT standard pour chaque bloc entraîne une oscillation spectrale très importante. Pour contourner ce problème, on propose une nouvelle approche de recherche d'image basée-région (RBIR) qui prend en considération une segmentation a priori de l'image en région [24]. Ceci dans l'objectif d'améliorer la similarité à la granularité de la région indépendamment des autres régions (ou l'arrière-plan). Ceci permet une recherche indépendante des objets individuels ce qui offre une nouvelle fonctionnalité.

3.3.1 Indexation d'image basée-objet utilisant Shape-Adaptive DCT

Dans notre système (Voir Fig.3.2), toutes les images de la base de données sont traitées en off-line. Premièrement, le système segmente chaque image en deux objets : premier-plan (l'objet principal) et arrière-plan puis applique la DCT 4x4 sur les blocs intérieurs et la SA-DCT sur les segments des blocs frontières de l'image. Une étape de pré-traitement est nécessaire pour normaliser le contraste et quantifier les blocs DCT et SA-DCT voir (Sec.3.3.4). Après l'étape du pré-traitement, les caractéristiques de texture et de couleur de bas niveau pour chaque région, AC-patterns et DC-patterns, sont extraites. Une fois les caractéristiques sont extraites, on désigne deux systèmes différents pour tester l'efficacité de la recherche. Ces deux systèmes diffèrent dans la formulation de la requête et le calcul de la mesure de similarité entre les images. Le premier système utilise la région d'intérêt comme requête [25], on le nomme : *Technique basée sur une seule région (région par région)*. Dans ce système, la distance entre la requête et les images de la base de données est le minimum des distances entre la région requête et toutes les régions des images de la base de données. Le deuxième système utilise une requête globale, comme SIMPLIcity [77], dans lequel une image requête entière est comparée avec les images entières de la base de données. On nomme ce système : *Technique basée sur l'image globale (image par image)*. Les

distances individuelles entre les régions des deux images sont combinées pour avoir une seule mesure de distance.

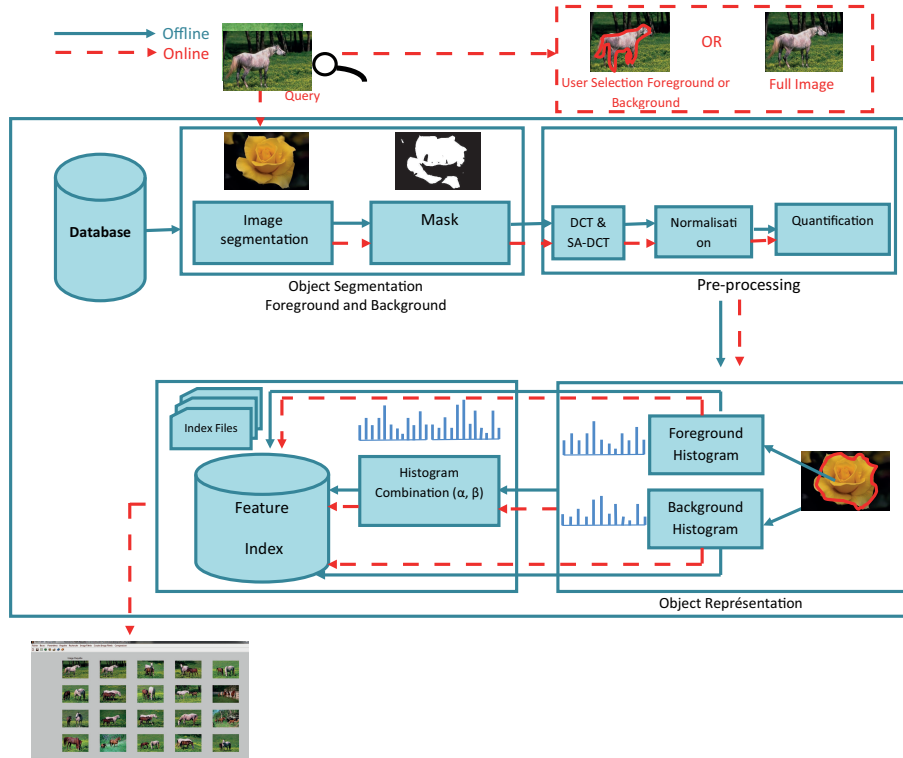


Figure 3.2 – Schéma synoptique de la méthode basée-région proposée

Pendant la recherche qui se fait en mode on-line, l'utilisateur sélectionne la région d'intérêt de l'image requête comme une région requête. On suppose que chaque image a au moins une région dominante (premier-plan) qui exprime sa sémantique. Le système calcule les caractéristiques de bas niveau de la région requête. Ensuite, un ensemble de N images est récupéré de la base de données. Cet ensemble contient les images qui ont des régions avec le même contenu que la région requête. Ces N régions sont classées selon leurs distances (eq.2.5) avec l'image requête. Le descripteur est basé sur l'histogramme combiné (eq. 2.2) (voir Fig. 2.6), du AC-patterns et DC-patterns, pour le premier-plan seulement, l'arrière-plan seulement ou bien pour l'image complète (premier-plan et l'arrière-plan) (Fig.3.3). Pour l'image complète, l'histogramme peut être combiné par la concaténation des histogrammes du premier-plan et de l'arrière-plan pour faire la recherche (voir Fig.3.5). Une concaténation générale est définie en appliquant β (eq. 3.1), un paramètre qui contrôle l'impact des deux histogrammes premier-plan, H_{Fore} , et arrière-plan, H_{Back} . Dans ce contexte, le descripteur global est défini comme suit :

$$H_{Full} = [(1 - \beta) \times H_{Fore}, \beta \times H_{Back}] \quad (3.1)$$

Le descripteur basé sur l'histogramme est motivé par l'importance de l'interrelation au long des coefficients DCT, les régions sont procédés de la même façon que chaque

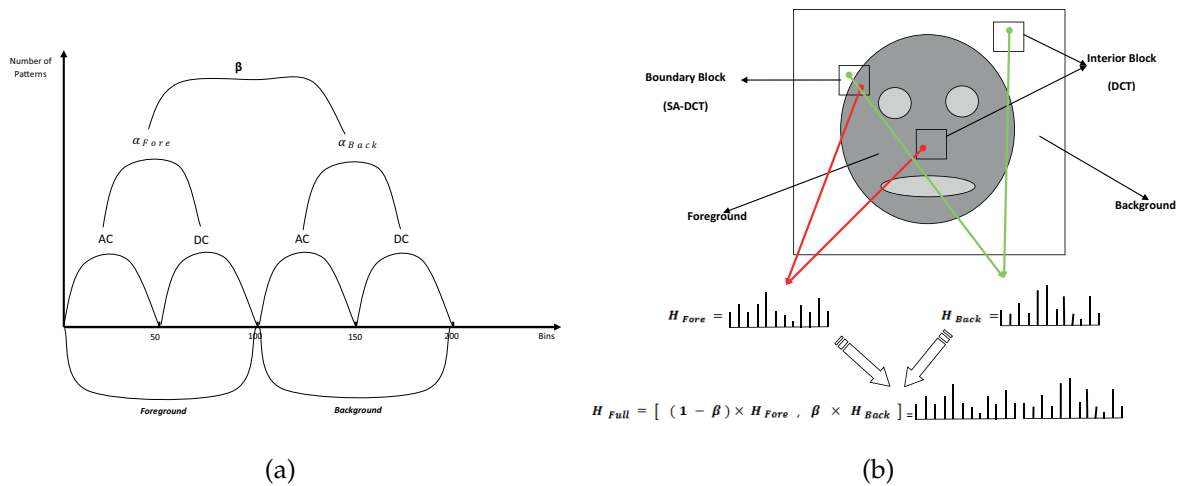


Figure 3.3 – Processus de construction de l'histogramme combiné par la méthode proposée. (a) Histogramme combiné en tenant compte des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$. (b) Illustration schématique montrant une image composée de deux objets, premier-plan et arrière-plan. Pour chaque objet, les blocs intérieurs sont transformés par la DCT, tandis que les segments des blocs frontières sont transformés par l'utilisation de la SA-DCT. H_{Fore} est construit à partir des blocs intérieurs et des segments des blocs frontières du premier-plan, alors que H_{Back} est construit à partir des blocs intérieurs et des segments des blocs frontières de l'arrière-plan. Une concaténation générale est définie en appliquant β pour combiner l'histogramme premier-plan H_{Fore} avec l'histogramme arrière-plan H_{Back} .

coefficient (SA-DCT) donne une partie des informations concernant la forme et le contenu couleur des objets.

La figure (Fig.3.4) représente les histogrammes des 50 AC-Patterns des premiers-plans des images Fleur et Cheval calculés par la méthode proposée. Tandis que la figure (Fig.3.5) représente les histogrammes combinés (premier-plan et arrière-plan) des images Fleur et Cheval calculés aussi par la méthode proposée.

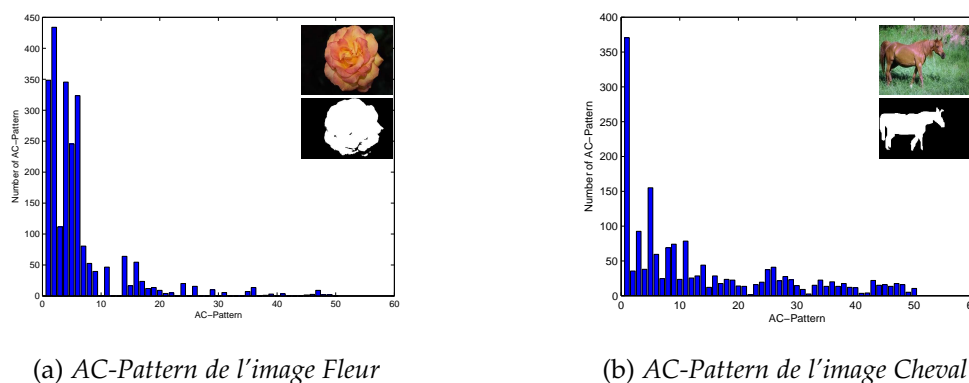
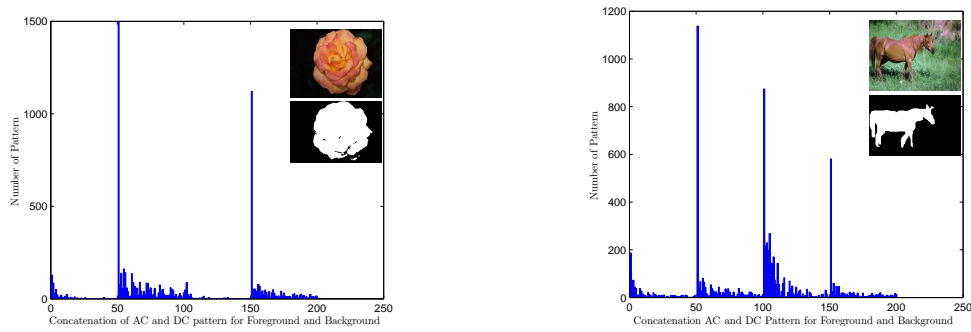


Figure 3.4 – Histogramme des 50 AC-patterns, hautes fréquences des occurrences de l'image (RBIR) (Region-Based Image Retrieval).



(a) Histogramme combiné de l'image Fleur

(b) Histogramme combiné de l'image Cheval

Figure 3.5 – Histogramme combiné des premières occurrences de hautes fréquences du AC-pattern et DC-pattern pour le RBIR (Region-Based Image Retrieval).

3.3.2 Segmentation d'image

Dans cette recherche, une technique de segmentation supervisée nommée Total Variation segmentation method (TV-Seg)[24] est adoptée. On utilise cette technique pour segmenter les images en prenant en compte la qualité de la segmentation et la complexité de calcul.

Le modèle des contour/snake actifs est l'un des modèles les plus connus dans la segmentation d'images. Il se compose d'une évolution de contours des images vers les frontières d'objets. Son succès est dû aux propriétés mathématiques solides et aux schémas numériques efficaces basés sur la méthode de courbes à niveau (level set). Le seul inconvénient de ce modèle est l'existence de minima locaux de l'énergie de contour actif, ce qui rend l'estimation initiale critique pour obtenir des résultats satisfaisants. Dans [24], ils proposent de résoudre ce problème en déterminant un minimum global du modèle de contour actif. Cette approche est basée sur l'unification de la segmentation d'image et la tâche de dé-bruitage d'image dans un cadre de minimisation globale. Plus précisément, ils proposent d'unifier trois modèles de variation d'images bien connus, à savoir le modèle de contours, le modèle de débruitage de Rudin-Sher-Fatemi et le modèle de segmentation de Mumford-Shah. Ils ont établi des théorèmes avec des preuves pour déterminer l'existence d'un minimum global du modèle de contour actif. D'un point de vue numérique, ils proposent un nouveau moyen pratique pour résoudre le problème de propagation de contour actif vers les frontières de l'objet à travers une double formulation du problème de minimisation. La formulation double, facile à implémenter, permet une minimisation globale rapide de l'énergie des snake. Elle permet d'éviter l'inconvénient majeur dans l'approche des courbes à niveau qui se compose d'initialisation du contour actif en fonction de la distance et re-initialisation périodiquement au cours de l'évolution, ce qui prend beaucoup de temps. [24] appliquent leurs algorithmes de segmentation sur les images synthétiques et naturelles, tels que les images de texture et les images médicales. Les résultats expérimentaux montrent la performance de ce modèle par rapport aux autres modèles de segmentation.

Il est noté qu'il existe d'autres méthodes de segmentation, comme [93, 92, 94], qui peuvent être utilisées dans notre système RBIR. Notons que l'objectif est l'étude de l'efficacité de la recherche des différentes représentations et non pas les algorithmes de segmentation.

3.3.3 Transformation adaptée aux objets SA-DCT

Après la segmentation des images, on a besoin d'extraire les caractéristiques des régions. On propose d'appliquer la DCT adaptée aux objets (SA-DCT) [71, 70] aux segments \mathcal{S} des blocs frontières de la région d'intérêt, et la DCT classique pour les blocs intérieurs. Le concept de base de la SA-DCT est d'appliquer premièrement la DCT 1-D verticale pour les pixels actifs, puis appliquer la DCT 1-D horizontale pour les coefficients de la DCT verticale avec le même indice de fréquence. Les détails spécifiques de cette transformation peuvent être vus dans la Partie I de cette thèse.

Après l'application de la DCT et la SA-DCT, la même démarche est appliquée pour extraire les textures-patterns (AC-Pattern) et les colors-patterns (DC-Pattern) comme il a été démontré dans la section (Sec.2.3).

3.3.4 Pré-traitement

Dans [86] et [88], l'auteur a proposé l'utilisation de l'histogramme des blocs DCT quantifiés pour la recherche des images.

Dans le système proposé, l'image est premièrement transformée en blocs DCT 4x4 pour les blocs intérieurs et en blocs SA-DCT pour les segments des blocs frontières. Les niveaux différents de luminance sont dus aux différents blocs DCT et SA-DCT. Pour normaliser la luminance, une étape de pré-traitement est appliquée après l'extraction des coefficients (DCT et SA-DCT) de bas niveau. Ceci est réalisé par la normalisation des coefficients par rapport à la moyenne de la luminance, qui est calculée à base des coefficients DC des blocs DCT et SA-DCT.

Supposant qu'il existe N blocs DCT et SA-DCT dans l'image i et que la valeur DC pour chaque bloc est dénotée par $DC_j(i)$, $1 \leq j \leq N$. À partir de ces valeurs DC, on peut calculer la valeur moyenne des DC pour cette image :

$$DC_{mean}(i) = \frac{1}{N} \sum_{j=1}^N DC_j(i) \quad (3.2)$$

Ensuite, la luminance moyenne $DC_{allmean}$ pour tous les M images de la base de données est calculée :

$$DC_{allmean} = \frac{1}{M} \sum_{i=1}^M DC_{mean}(i) \quad (3.3)$$

Puis le facteur d'échelle du changement de luminance pour l'image i est calculé par :

$$R_i = \frac{DC_{allmean}}{DC_{mean}(i)} \quad (3.4)$$

Et finalement, tous les coefficients (AC et DC) de la DCT et la SA-DCT de l'image i sont normalisés, par rapport à R_i , par leur facteur d'échelle :

$$\widehat{DCT}_i = DCT_i \times R_i, \quad (3.5)$$

$$\widehat{SADCT}_i = SADCT_i \times R_i \quad (3.6)$$

Où DCT_i sont les coefficients DCT (AC et DC) de l'image i issus des blocs intérieurs et $SADCT_i$ sont les coefficients SA-DCT (AC et DC) de l'image i issus des segments des blocs frontières.

Après cette normalisation, les coefficients AC et DC des blocs DCT et SA-DCT sont quantifiés par deux paramètres de quantification QP_{AC} et QP_{DC} respectivement :

$$\overline{AC}_i = \frac{\widehat{AC}_i}{QP_{AC}}, \quad (3.7)$$

$$\overline{DC}_i = \frac{\widehat{DC}_i}{QP_{DC}} \quad (3.8)$$

Où \widehat{AC}_i et \widehat{DC}_i sont les coefficients AC_i et DC_i des blocs DCT et SA-DCT après une normalisation.

Dans l'approche proposée, la construction des AC-patterns et DC-patterns (Sec.2.3.1, Sec.2.3.2), issus des blocs DCT et SA-DCT, se fait après l'étape du pré-traitement.

3.3.5 Mesure de similarité

Un des problèmes d'un système de recherche d'images par objet est comment comparer entre deux images, c'est-à-dire la définition de mesure de similarité entre les images. Une simple solution adoptée par les premiers systèmes [58], [25] est l'utilisation de mesure similarité individuelle *région par région*. Pour utiliser un tel système, l'utilisateur est censé de sélectionner une ou plusieurs régions à partir de l'image requête pour faire la recherche. Comme il a été discuté dans [77], dû à la nature incontrôlable des images, extraire les objets d'images automatiquement et précisément est encore hors de portée de l'état de l'art des nouvelles techniques de vision par ordinateur. Toutefois, d'autres systèmes ont tendance à partitionner un objet en plusieurs régions, dont aucune est représentative de l'objet sémantique. Par conséquent, il est souvent difficile pour les utilisateurs de déterminer quelles régions devraient être utilisées pour la recherche. Pour fournir aux utilisateurs une interface plus simple et réduire l'influence de la segmentation inexacte, des mesures de similarités *image par image* qui combinent les propriétés de toutes les régions ont été proposées dans [77],[8]. Ces systèmes exigent seulement aux utilisateurs de choisir la requête, et donc les libèrent d'énigmatiques décisions sur les régions. Par exemple, le système SIMPLYcity [77] utilise IRM (Integrated Region Matching) comme mesure de similarité. En permettant une relation *plusieurs à plusieurs* entre les régions. L'approche est robuste contre les segmentations inexactes.

Dans les définitions des mesures de similarité *image par image*, le facteur important est le poids des régions. Ce problème est généralement réglé par l'utilisation des informations préalables. Par exemple, dans [77],[8], le pourcentage de la région est utilisé pour décider l'importance de la région.

Dans le système proposé, deux mesures de similarité sont adoptées, *région par région* et *image par image*, selon que la recherche est par région ou bien par image complète.

3.3.5.1 Mesure de similarité *image par image*

Fondées sur l'hypothèse que toute région pourrait être utile lors de l'évaluation de la similarité [77], toutes les régions de chaque image sont considérées. Tout d'abord, il faut noté que le calcul de similarité, adopté par notre système, entre une région d'une image et une autre région d'une autre image est le $\chi^{2\text{fin}}$ (éq.2.5).

Le calcul de similarité entre les images est basé sur le calcul de toutes les similarités entre les régions après segmentation puis de prendre la somme des minimums entre les distances des régions. Le tableau (Tab.3.1) et la figure (Fig.3.6) montrent un exemple du

processus pour calculer la similarité entre deux images.

B/R	Premier-plan	Arrière-plan
Premier-plan	0.2	0.7
Arrière-plan	0.3	0.5
Min	0.2	0.5

Table 3.1 – Similarité entre les régions

Après la construction du tableau (Tab.3.1), on calcule la somme des valeurs de la dernière ligne pour obtenir la similarité entre deux images.

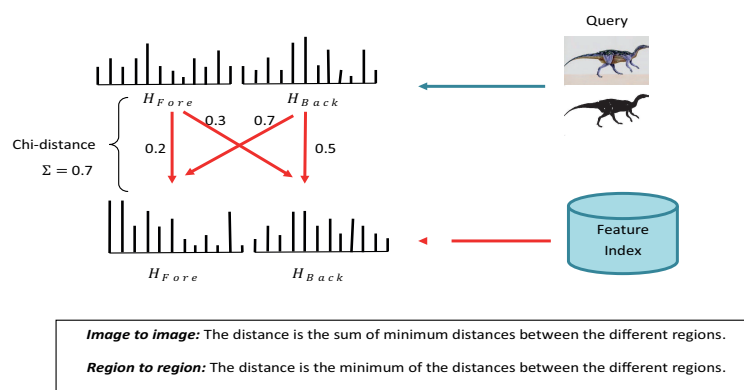


Figure 3.6 – Exemple de calcul de similarité entre les images

3.3.5.2 Mesure de similarité région par région

La deuxième possibilité adoptée par notre système est le calcul de similarité *région par région*. L'utilisateur sélectionne une région parmi les deux régions possibles (premier-plan ou arrière-plan) à partir de l'image requête pour faire la recherche. La mesure de similarité *région par région* consiste à calculer la similarité entre la région sélectionnée et les régions des images de la base de données puis prendre le minimum des distances entre les régions.

3.3.6 Résumé

La figure (Fig3.7) représente un récapitulatif sous forme d'un schéma blocs général du système proposé. Chaque bloc dans ce schéma a été détaillé auparavant.

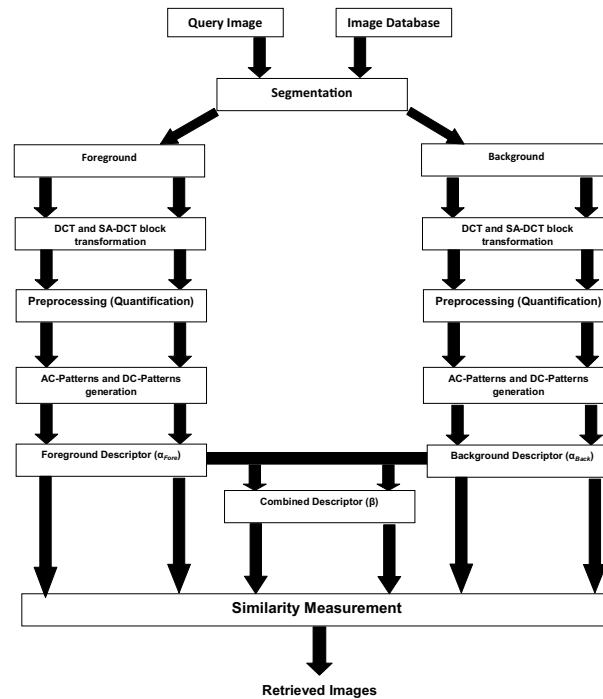


Figure 3.7 – Schéma blocs de la méthode RBIR proposée utilisant la transformation SA-DCT

3.4 RÉSULTATS EXPÉRIMENTAUX

Pour tester l'efficacité des systèmes proposés, plusieurs tests ont été effectués sur des images de bases différentes : Corel-1000, Caltech-256 et GTF. La base de données Corel-1000 a été collecté par Wang et al.[30], contient 1000 256x384 ou 384x256 images, ces images sont classifiées en 10 catégories sémantiques (voir Table. 3.2) : Africains, bâtiments, plages, bus, dinosaures, éléphants, fleurs, chevaies, montagnes et plats.

La base de données Caltech-256 a été collecté par Griffin et al. [42], [6], contient 30607

ID	Catégories
1	Africains
2	Plages
3	Bâtiments
4	Bus
5	Dinosaures
6	Éléphants
7	Fleurs
8	Chevaies
9	Montagnes
10	Plats

Table 3.2 – Les catégories de Corel-1000 des images testées

images de 256 catégories, 80 à 827 images par catégorie. Dans nos expérimentations, on a sélectionné aléatoirement 10 catégories (voir Table. 3.3) à partir de Caltech-256, qui contiennent 1299 images. Les 10 catégories sont : Ak-47, Drapeaux américains, Sacs-à-dos, Battes-baseball, Gants-baseball, Cercles-basketball, Chauves-souris, Baignoires, Chopes à bière et Ballons.

ID	Catégories
1	Ak-47
2	Drapeau américain
3	Sac-à-dos
4	Batte-baseball
5	Gant-baseball
6	Cercle-basketball
7	Chauve-souris
8	Baignoire
9	Chope à bière
10	Ballon

Table 3.3 – Les catégories de Caltech-256 des images testées

La base de donnée GTF contient 15 images de 50 personne [43]. Les images sont de résolution 640x640 pixels tel que la taille de l'image est 150x150 pixels.

3.4.1 Coût de calculs

Les expérimentations dans cette partie sont implémentées sur une machine Intel (R) Xeon (R) Quad-Core X5650 (2.67 GHz and 2.66 GHz, deux processeurs, 12 GB RAM), système d'exploitation 64 bit Windows 7, environnement MATLAB R2009. On considère le scénario classique de recherche, là où le calcul du vecteur de caractéristiques pour tous les images dans la base de données se fait en off-line et le calcul du vecteur de caractéristiques de l'image requête se fait en on-line. On mesure le temps moyen d'exécution pour chaque requête, en général, une seconde est nécessaire pour segmenter une image. Notons que le temps nécessaire pour la représentation de caractéristiques incluant les transformations DCT et SADCT, la construction du vecteur de caractéristique et de l'histogramme est à peu près 36 secondes par image. La recherche est vraiment rapide. Quand l'image requête est dans la base de données, ça prend 1 seconde de temps CPU pour trier et ordonner toutes les images de la base de données par l'utilisation de la mesure de similarité Chi-square (éq.2.5).

3.4.2 Ajustement de paramètres

Dans le système proposé, on distingue deux types de paramètres qui peuvent être ajustés. Les paramètres dédiés à l'indexation et les paramètres dédiés à la recherche.

3.4.2.1 Paramètres d'indexation

Il existe plusieurs paramètres d'indexation, décrits dans les chapitres 2 et 3, utilisés dans le système proposé. Ces paramètres doivent être optimisés pour améliorer la performance du système.

- QP_{AC} et QP_{DC} : **Paramètres de quantification pour le AC et le DC respectivement.** Si ces paramètres sont assez petits, le nombre total des différents AC-patterns et DC-patterns devient plus grand, ceci met le processus de génération de l'histogramme plus compliqué et plus couteux en temps. Par contre, si les paramètres QP_{AC} et QP_{DC} sont aussi grands, alors les coefficients AC-patterns et DC-patterns les plus à droite tendent vers zero, ceci décroît la performance de la recherche. Donc, il faut trouver un compromis pour le choix de ces paramètres entre la performance et la consommation du temps voir (Sec.3.3.4).
- γ : **premières directions avec les plus grandes différences DC-Pattern.** Ce paramètre peut être ajusté pour avoir une meilleure performance de recherche voir (Sec.2.3.2).
- ACBins et DCBins : **Nombres des AC bins et DC bins nécessaires pour construire l'histogramme de caractéristiques.**

Les histogrammes sont construits par l'ordonnement des occurrences de caractéristiques. L'histogramme résultant a des bins ordonnés dans un ordre décroissant. La taille de cet histogramme est un paramètre libre qui peut être ajusté pour améliorer la performance de la recherche. L'élimination de quelques bins qui représentent des caractéristiques non-pertinentes améliore la performance de la recherche. Par contre, l'élimination de plusieurs bins en résulte la dégradation de performances. Alors il faut choisir la longueur optimale qui induit une meilleure recherche.

- **NC : Le nombre des coefficients AC entrant dans la construction du AC-pattern.** NC représente le nombre de coefficients AC qui forment les groupes diagonaux, horizontales et verticales vues dans la section (Sec.2.3.1). Ce paramètre peut prendre la valeur $NC = 2$ ou bien $NC = 3$.

La figure (Fig.3.8) illustre le processus d'apprentissage permettant d'optimiser les paramètres d'indexation cités en haut.

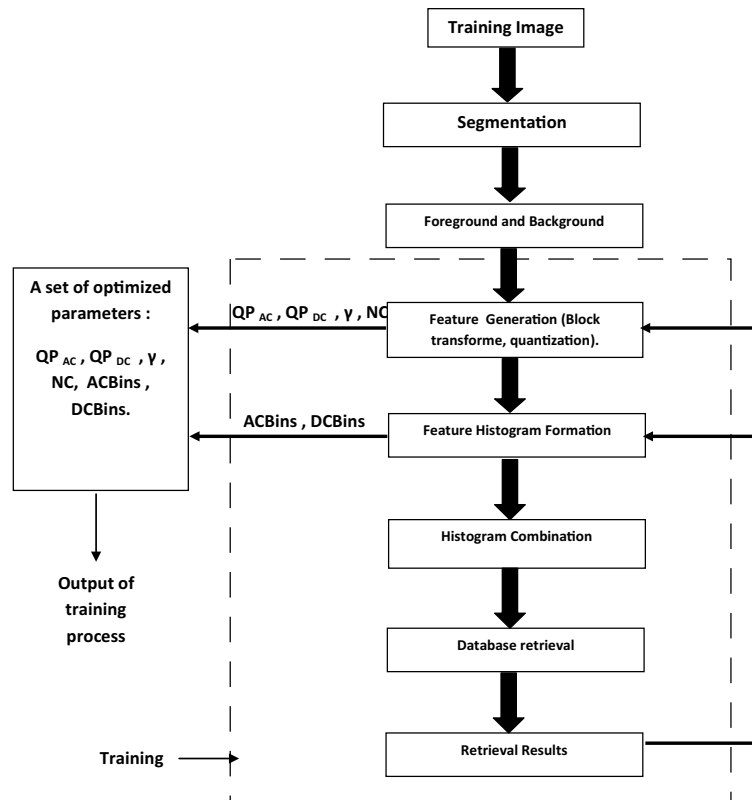


Figure 3.8 – Processus d'apprentissage optimisant les paramètres d'indexation QP_{AC} , QP_{DC} , γ , NC , $ACBins$, $DCBins$.

3.4.2.2 Paramètres de la recherche

Il existe plusieurs paramètres de la recherche utilisés dans le système proposé. Ces paramètres doivent être aussi optimisés pour améliorer la performance du système.

- β : un poids qui combine l'histogramme du premier-plan avec celui de l'arrière-plan.

L'équation (éq.3.1) représente l'histogramme combiné du premier-plan et de l'arrière-plan en utilisant un paramètre poids β qui contrôle l'impact du premier-plan sur l'arrière-plan. Ce paramètre peut être ajusté pour améliorer la performance du système proposé.

- α_{Fore} : un poids qui combine l'histogramme AC avec celui du DC pour le premier-plan.

L'équation (éq.3.9) représente l'histogramme combiné du premier-plan H_{Fore} ; où α_{Fore} est un paramètre poids qui contrôle l'impact de l'histogramme AC sur celui du DC du premier-plan. Ce paramètre aussi, peut être ajusté pour améliorer la qualité de la recherche.

$$H_{Fore} = [(1 - \alpha_{Fore}) \times H_{AC_{Fore}}, \alpha_{Fore} \times H_{DC_{Fore}}] \quad (3.9)$$

- α_{Back} : un poids qui combine l'histogramme AC avec celui du DC pour l'arrière-plan.

L'équation (éq.3.10) représente l'histogramme combiné de l'arrière-plan H_{Back} ; où α_{Back} est un paramètre poids qui contrôle l'impact de l'histogramme AC sur celui du DC de l'arrière-plan. Ce paramètre aussi, peut-être ajusté pour améliorer la qualité de la recherche.

$$H_{Back} = [(1 - \alpha_{Back}) \times H_{AC_{Back}}, \alpha_{Back} \times H_{DC_{Back}}] \quad (3.10)$$

La figure (Fig.3.9) illustre le processus d'apprentissage permettant d'optimiser les paramètres de la recherche $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$.

Démonstrations : La substitution de (éq.3.9) et (éq.3.10) dans (éq. 3.1) donne l'équation suivante :

$$H_{Full} = [(1 - \alpha_{Fore} - \beta + \alpha_{Fore}\beta) \times H_{AC_{Fore}}, (\alpha_{Fore} - \alpha_{Fore}\beta)H_{DC_{Fore}}, (\beta - \beta\alpha_{Back}) \times H_{AC_{Back}}, \beta\alpha_{Back} \times H_{DC_{Back}}] \quad (3.11)$$

Plusieurs cas peuvent apparaître :

1. Si $\alpha_{Fore} = 1$ et $\alpha_{Back} = 1 \Rightarrow H_{Full} = [(1 - \beta) \times H_{DC_{Fore}}, \beta H_{DC_{Back}}]$. On aura une recherche avec le DC du premier-plan et de l'arrière-plan.
 Si $\beta = 1 \Rightarrow H_{Full} = H_{DC_{Back}}$, une recherche avec le DC de l'arrière-plan.
 Si $\beta = 0 \Rightarrow H_{Full} = H_{DC_{Fore}}$, une recherche avec le DC du premier-plan.

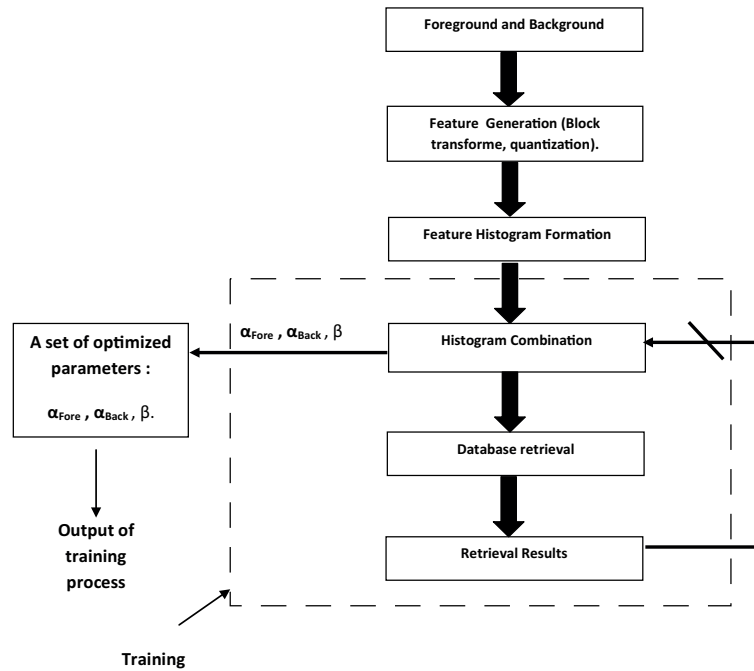


Figure 3.9 – Processus d'apprentissage optimisant les paramètres de la recherche $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$.

- 2. Si $\alpha_{Fore} = 0$ et $\alpha_{Back} = 0 \Rightarrow H_{Full} = [(1 - \beta) \times H_{AC_{Fore}}, \beta H_{AC_{Back}}]$. On aura une recherche avec le AC du premier-plan et de l'arrière-plan.
 Si $\beta = 1 \Rightarrow H_{Full} = H_{AC_{Back}}$ une recherche avec le AC de l'arrière-plan.
 Si $\beta = 0 \Rightarrow H_{Full} = H_{AC_{Fore}}$ une recherche avec le AC du premier-plan.
- 3. Si $\alpha_{Fore} = \alpha_{Back} = \beta = \frac{1}{2} \Rightarrow H_{Full} = [\frac{1}{4}H_{AC_{Fore}}, \frac{1}{4}H_{DC_{Fore}}, \frac{1}{4}H_{AC_{Back}}, \frac{1}{4}H_{DC_{Back}}]$

Le tableau (Tab.3.4) montre un exemple de combinaison des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ lorsqu'il s'agit de valeurs binaires. D'après les résultats trouvés, on remarque qu'il

Combinaison	α_{Fore}	α_{Back}	β	H_{Full}
1	0	0	0	$H_{AC_{Fore}}$
2	0	0	1	$H_{AC_{Back}}$
3	0	1	0	$H_{AC_{Fore}}$
4	0	1	1	$H_{DC_{Back}}$
5	1	0	0	$H_{DC_{Fore}}$
6	1	0	1	$H_{AC_{Back}}$
7	1	1	0	$H_{DC_{Fore}}$
8	1	1	1	$H_{DC_{Back}}$

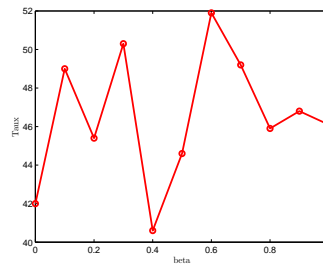
Table 3.4 – H_{Full} pour les combinaisons binaires des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$.

existe des combinaisons de $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ qui donnent le même H_{Full} . Il s'agit des combinaisons 1 et 3, 2 et 6, 4 et 8 et enfin 5 et 7.

La variation des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ implique une variation de type de recherche, ce qui permet de choisir les paramètres les mieux adaptés pour chaque base de données. En plus, le système RBIR proposé offre plus de souplesse en comparaison avec le système

CBIR classique. Le système classique (CBIR avec DCT seulement) permet la variation seulement d'une seule valeur α qui contrôle l'impact du AC sur le DC, alors le système RBIR proposé permet la variation des trois paramètres, $\{ \alpha_{Back}, \alpha_{Fore}, \beta \}$, ceci permet d'améliorer la recherche.

La courbe de la figure (Fig.3.10), illustre le MAP (éq.2.9) pour la base de données Corel-1000 en fonction de la variation de β .



(a)

Figure 3.10 – MAP en fonction de la variation du paramètre β pour la base de données Corel-1000

On peut aller plus loin, les paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ peuvent être choisis pour chaque classe, cela veut dire, avoir pour chaque base de données non seulement une seule valeur de α_{Fore} , α_{Back} et β mais un vecteur de valeurs de $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ selon le nombre de classes. Les expérimentations ont montré que ceci améliore d'une manière très attrayante la qualité de la recherche.

Pour choisir les paramètres optimaux $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$, on les fait varier et on choisit les valeurs qui donnent le meilleur MAP (éq.2.9) pour chaque classe. Les courbes de la figure (Fig.3.11) montrent le MAP en fonction de la variation de β pour chaque classe de la base de données Corel-1000. Pour la classe2, par exemple, on doit prendre $\beta = 0.1$ pour avoir un MAP qui vaut 100%, alors que pour la classe9, β doit prendre la valeur 0.7 pour aboutir à un MAP qui est égal à 100%. Par conséquent, améliorer le MAP pour chaque classe implique son amélioration pour toute la base de données.

Mais le problème qui se pose est si on veut faire une recherche avec une image qui ne se trouve pas dans la base de données, on doit choisir une classe pour cette image pour lui attribuer la valeur des $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ appropriée. Cependant, pour un simple utilisateur du logiciel cela n'est pas évident car son objectif est de faire seulement la recherche et non pas de modifier les paramètres du logiciel.

Il existe des systèmes comme [4] et [5], qui n'autorisent pas la recherche des images hors de la base de données, mais ils offrent aux utilisateurs le choix de l'image requête parmi celles qui se trouvent dans la base. On peut classer notre système là où les paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ sont variables pour chaque classe parmi les systèmes qui n'autorisent pas

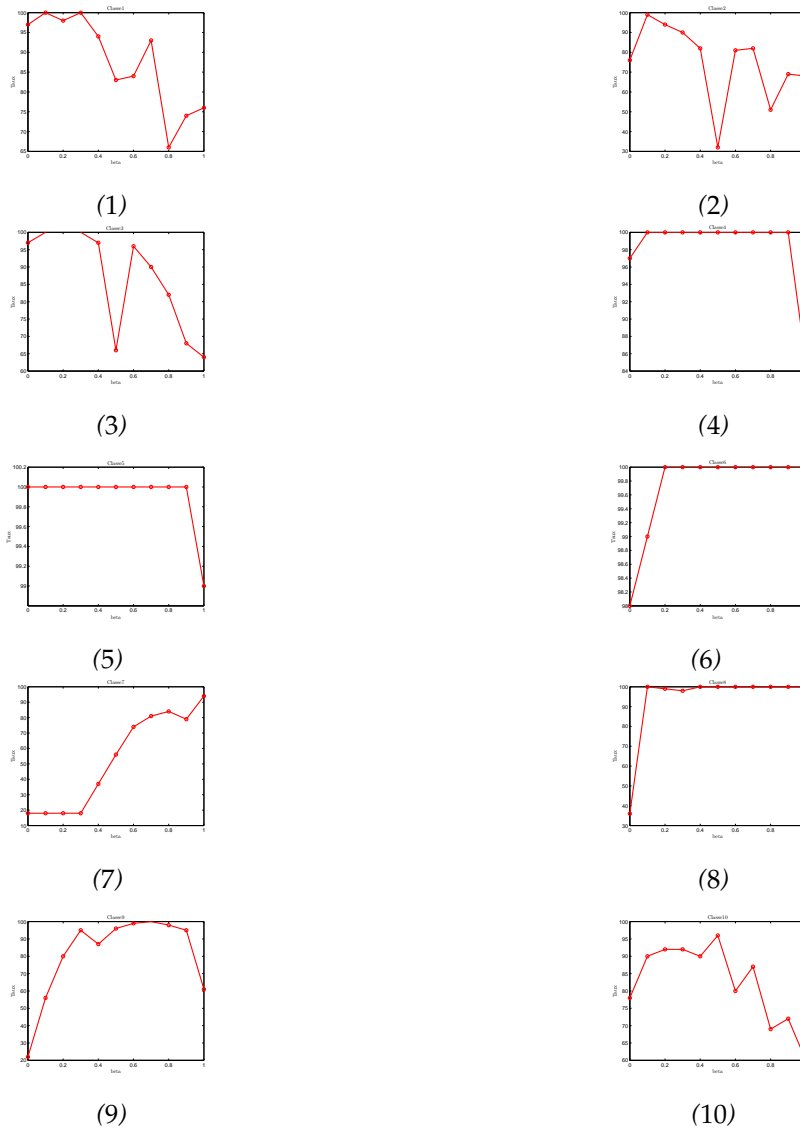


Figure 3.11 – MAP en fonction de la variation du paramètre β pour chaque classe : les courbes (1)-(10) sont pour les classes de la base de données Corel-1000

la recherche d'images hors de la base de données.

Par la variation des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ pour toute la base de données ou bien pour chaque classe, deux systèmes différents peuvent apparaître. Le premier système est celui qui prend une seule valeur de $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ pour toute la base, on le nomme *Système avec paramètres globaux*. Le deuxième système où les paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ sont choisis pour chaque classe, on le nomme *Système avec paramètres adaptés pour chaque classe* (Voir Fig.3.12).

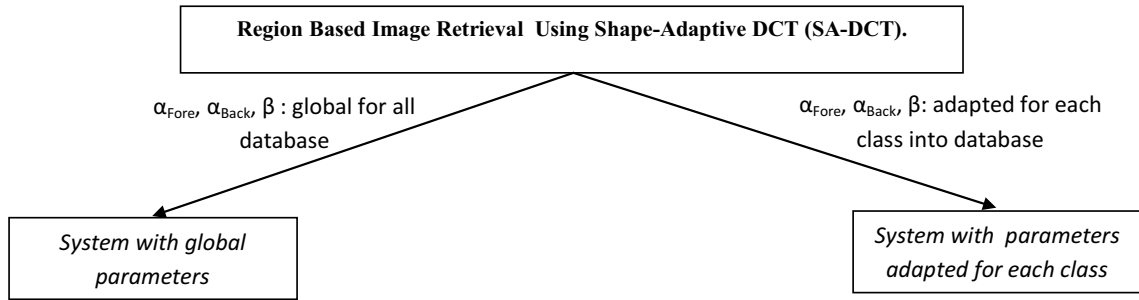


Figure 3.12 – schéma d’illustration montrant deux Systèmes résultants par la variation des paramètres : Système avec paramètres globaux et Système avec paramètres adaptés pour chaque classe.

3.4.3 Optimisation

Dans la section suivante, on présente une manière pour optimiser notre recherche. Elle consiste à séparer les AC-patterns et DC-patterns de l’objet en patterns frontières et patterns intérieurs tels qu’il est montré dans la figure (Fig.3.13).

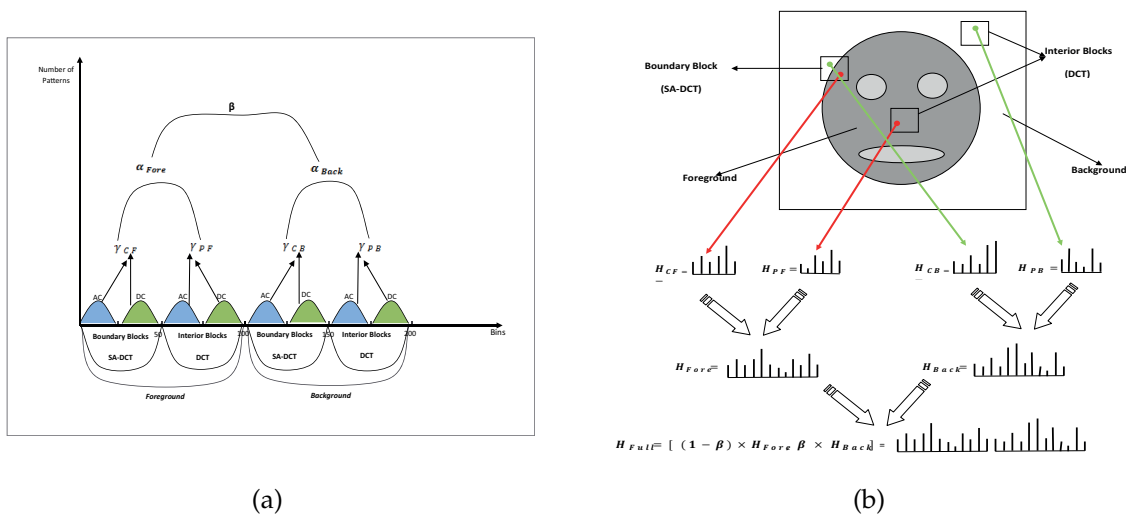


Figure 3.13 – Processus de construction de l’histogramme combiné par la méthode proposée : principe de l’optimisation de la recherche. (a) Histogramme combiné en tenant compte des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ et $\{ \gamma_{CF}, \gamma_{PF}, \gamma_{CB}, \gamma_{PB} \}$. (b) Schéma d’illustration montrant une image composée de deux objets, premier-plan et arrière-plan. L’histogramme du premier-plan, H_{Fore} , est combiné à partir de l’histogramme issu des blocs intérieurs, H_{PF} , et l’histogramme issu des segments de blocs frontières, H_{CF} , du premier-plan. Alors que l’histogramme de l’arrière-plan, H_{Back} , est combiné à partir de l’histogramme issu des blocs intérieurs, H_{PB} , et l’histogramme issu des segments de blocs frontières, H_{CB} , de l’arrière-plan. L’histogramme combiné global est calculé à partir des histogrammes premier-plan, H_{Fore} , et arrière-plan, H_{Back} .

Donc , en plus des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$, on a ajouté d’autres paramètres $\{ \gamma_{CF}, \gamma_{PF}, \gamma_{CB}, \gamma_{PB} \}$ qui permettent de gérer les poids entre les patterns issus des blocs intérieurs et des segments des blocs frontières de l’image. Les équations suivantes illustrent

l'emploi de ces paramètres pour la construction du descripteur de l'image.

1. *Pour le premier-plan*

— Frontière du premier-plan

$$H_{CF} = [(1 - \gamma_{CF}) \times H_{AC_{CF}}, \gamma_{CF} \times H_{DC_{CF}}] \quad (3.12)$$

H_{CF} : Histogramme combiné des patterns (AC-DC) issu des segments des blocs frontières.

$H_{AC_{CF}}$: Histogramme AC-pattern issu des segments des blocs frontières.

$H_{DC_{CF}}$: Histogramme DC-pattern issu des segments des blocs frontières.

γ_{CF} : Un poids pour combiner H_{CF} .

— Intérieur du premier-plan

$$H_{PF} = [(1 - \gamma_{PF}) \times H_{AC_{PF}}, \gamma_{PF} \times H_{DC_{PF}}] \quad (3.13)$$

H_{PF} : Histogramme combiné des patterns (AC-DC) issu des blocs intérieurs.

$H_{AC_{PF}}$: Histogramme AC-pattern issu des blocs intérieurs.

$H_{DC_{PF}}$: Histogramme DC-pattern issu des blocs intérieurs.

γ_{PF} : Un poids pour combiner H_{PF} .

— (Intérieur + Frontière) du premier-plan

$$H_{Fore} = [(1 - \alpha_{Fore}) \times H_{PF}, \alpha_{Fore} \times H_{CF}] \quad (3.14)$$

2. *Pour l'arrière-plan*

— Frontière de l'arrière-plan

$$H_{CB} = [(1 - \gamma_{CB}) \times H_{AC_{CB}}, \gamma_{CB} \times H_{DC_{CB}}] \quad (3.15)$$

H_{CB} : Histogramme combiné des patterns (AC-DC) issu des segments des blocs frontières.

$H_{AC_{CB}}$: Histogramme AC-pattern issu des segments des blocs frontières.

$H_{DC_{CB}}$: Histogramme DC-pattern issu des segments des blocs frontières.

γ_{CB} : Un poids pour combiner H_{CB} .

— Intérieur de l'arrière-plan

$$H_{PB} = [(1 - \gamma_{PB}) \times H_{AC_{PB}}, \gamma_{PB} \times H_{DC_{PB}}] \quad (3.16)$$

H_{PB} : Histogramme combiné des patterns (AC-DC) issu des blocs intérieurs.

$H_{AC_{PB}}$: Histogramme AC-pattern issu des blocs intérieurs.

$H_{DC_{PB}}$: Histogramme DC-pattern issu des blocs intérieurs.

γ_{PB} : Un poids pour combiner H_{PB} .

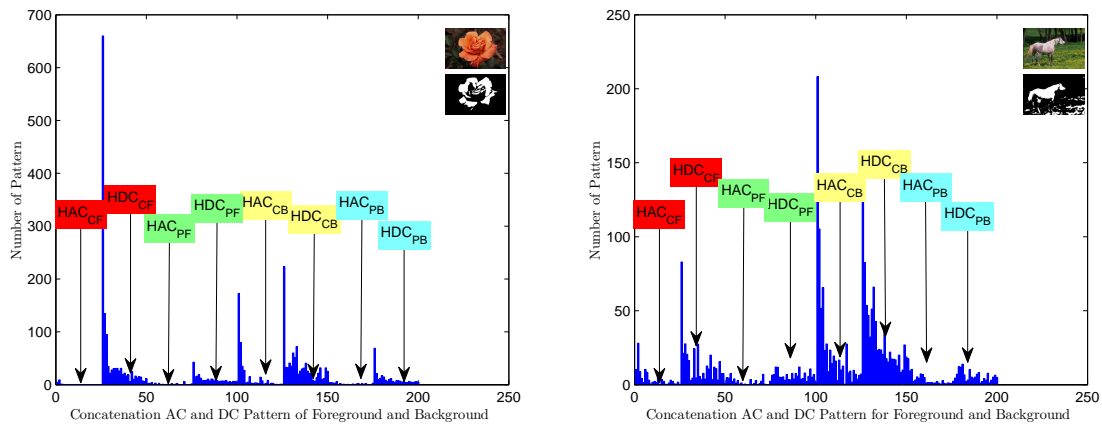
— (Intérieur + Frontière) arrière-plan

$$H_{Back} = [(1 - \alpha_{Back}) \times H_{PB}, \alpha_{Back} \times H_{CB}] \quad (3.17)$$

3. Pour l'image complète

$$H_{Full} = [(1 - \beta) \times H_{Fore}, \beta \times H_{Back}] \quad (3.18)$$

Pour l'image complète, l'histogramme peut être combiné par la concaténation des histogrammes du premier-plan et de l'arrière-plan pour faire la recherche. Les figures (Fig.3.14(a)) et (Fig.3.14(b)) représentent les descripteurs des images Fleur et Cheval respectivement, calculés par la méthode citée en haut.



(a) Histogramme combiné de l'image Fleur

(b) Histogramme combiné de l'image Cheval

Figure 3.14 – Histogramme combiné des premières occurrences de hautes fréquences du AC-Patterns et DC-Patterns pour le RBIR (Region-Based Image Retrieval) : Principe de l'optimisation.

3.4.4 Système avec paramètres globaux (Fig.3.12)

3.4.4.1 Résultats avec les bases de données Corel-1000 et Caltech-256

Pour évaluer la performance de notre système RBIR (DCT+SA-DCT) on le compare avec le système classique (CBIR avec DCT seulement), celui qui donne une meilleure précision pour le même rappel est le système le plus performant.

Les paramètres $\{ \beta, \alpha_{Fore}, \alpha_{Back}, \alpha(DCT) \}$ utilisés pour les expérimentations des deux bases de données Corel-1000 et Caltech-256 sont montrés dans les tableaux suivants (Tab.3.5) et (Tab.3.6) respectivement.

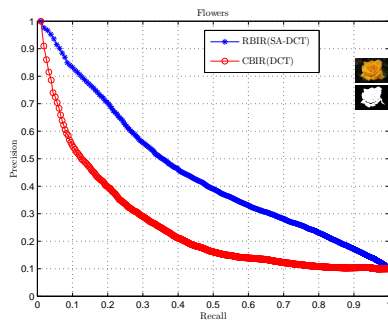
Paramètres	Valeurs
β	0.6
α_{Fore}	0.1
α_{Back}	0.1
$\alpha(DCT)$	0.7

Table 3.5 – Valeurs optimales de $\{ \beta, \alpha_{Fore}, \alpha_{Back}, \alpha(DCT) \}$ pour la base de données Corel-1000

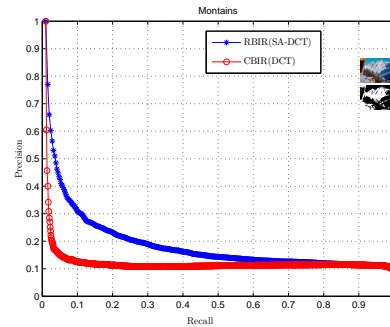
Paramètres	Valeurs
β	0.3
α_{Fore}	0.4
α_{Back}	0.4
$\alpha(DCT)$	0.5

Table 3.6 – Valeurs optimales de $\{ \beta, \alpha_{Fore}, \alpha_{Back}, \alpha(DCT) \}$ pour la base de données Caltech-256

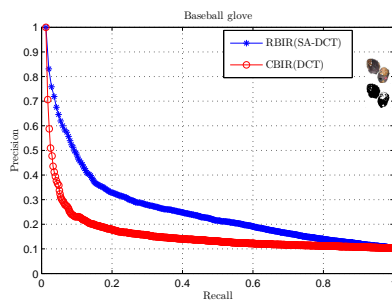
Dix catégories d'images sont testées. Les courbes moyennes de précision-rappel entre notre approche (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement) sont montrées dans la figure (Fig.3.15).



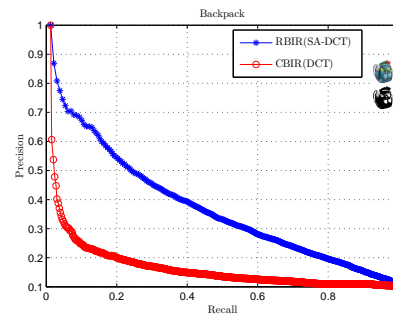
(a) P-R utilisant "Fleurs" de Corel-1000



(b) P-R utilisant "Montagnes" de Corel-1000



(c) P-R utilisant "Gants-Baseball" de Caltech-256



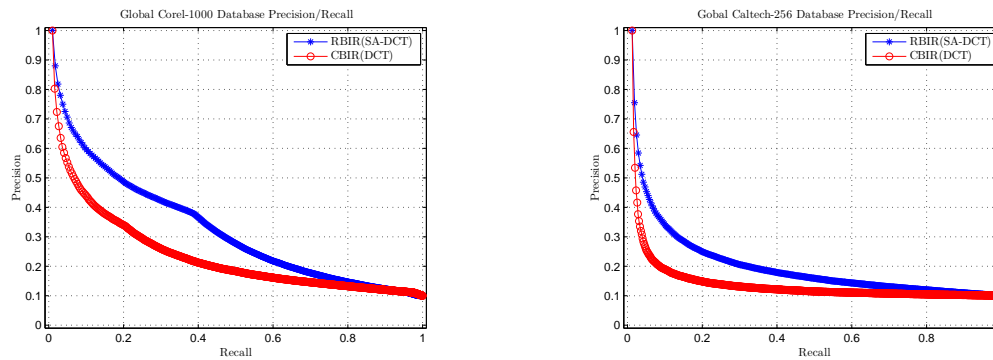
(d) P-R utilisant "Sacs-à-dos" de Caltech-256

Figure 3.15 – Les courbes précision-rappel moyennes entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement). Les catégories d'images (a) "Fleurs" (b) "Montagnes" sont de la base de données Corel-1000, et les catégories d'images (c) "Gants de baseball" (d) "Sacs-à-dos" sont de la base de données Caltech-256

La comparaison des résultats des courbes moyennes de précision-rappel, pour la base de données Corel-1000, entre l'approche proposée basée-région et l'approche conventionnelle basée sur le contenu [86] est montrée dans la figure (Fig.3.16(a)). Les mêmes courbes, pour la base de données Caltech-256, sont montrées dans la figure (Fig.3.16(b)). La meilleure performance est obtenue par l'utilisation du descripteur global combiné (eq. 3.1) pour le premier-plan et l'arrière-plan ensemble.

Pour la base de données Corel-1000, le système proposé permet de contourner les limitations du système classique (CBIR avec DCT seulement), ceci est dû à l'élimination de l'influence de l'arrière-plan par l'emploi de la SA-DCT.

Par contre pour la base de données Caltech-256, le système proposé n'a pas rapporté une grande amélioration par rapport au système classique (CBIR avec DCT seulement), car les catégories de cette base de données sont très rapprochées ce qui entraîne une confusion dans la recherche.



(a) P-R de la base de données Corel-1000

(b) P-R de la base de données Caltech-256

Figure 3.16 – Les courbes précision-rappel moyennes : (a) Base de données Corel-1000, (b) Base de données Caltech-256, entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement)[86]

3.4.4.2 Résultats avec la base de données visage GTF

Equal Error Rate (EER) [22](Sec.2.5.4), est souvent utilisé pour évaluer la performance d'un système de reconnaissance de visage. Un EER faible indique la performance du système.

Les paramètres nécessaires pour les expérimentations avec cette base de données sont montrés dans le tableau (Tab.3.7).

Paramètres	Valeurs
β	0.6
α_{Fore}	0.6
α_{Back}	1
$\alpha(DCT)$	0.6

Table 3.7 – Valeurs optimales de $\{ \beta, \alpha_{Fore}, \alpha_{Back}, \alpha(DCT) \}$ pour la base de données GTF

On utilise les descripteurs du AC-pattern et DC-pattern ensemble pour faire la reconnaissance du visage. Les paramètres α_{Fore} et α_{Back} voir (éq.3.9) et (éq.3.10) sont changés pour voir la performance d'une comparaison globale, comme montré dans la figure (Fig.3.18(a)) pour la base de données GTF. La meilleure performance, avec $EER = 0.076$, est obtenue par l'utilisation du descripteur combiné (éq.3.1) du premier-plan (visage) et de l'arrière-plan ensemble avec un $\beta = 0.6$. Les courbes précision-rappel dans la figure (Fig.3.18(b)) confirment les résultats obtenus par le EER . Donc, la méthode proposée améliore aussi la performance pour la base de données GTF. En outre, on a utilisé un nombre minimal de coefficients AC et un nombre faible de coefficients d'histogramme pour réduire le temps de calcul.

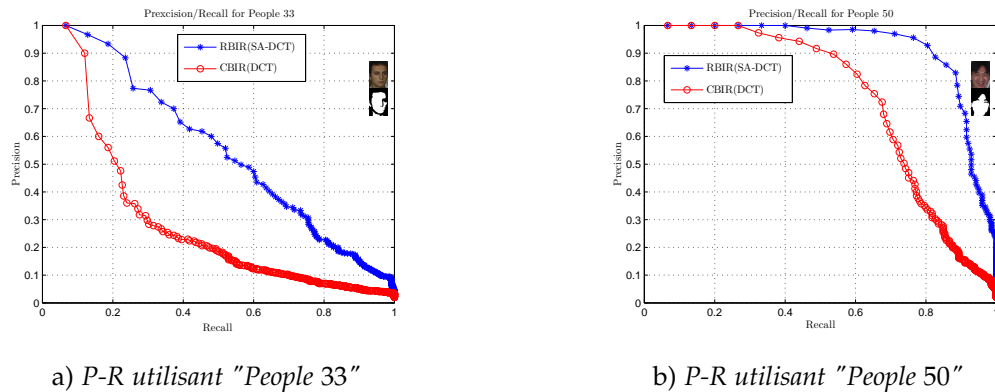


Figure 3.17 – Les courbes précision-rappel moyennes entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement). Les catégories d'images (a)"People 33" (b)"People 50" sont de la base de données GTF

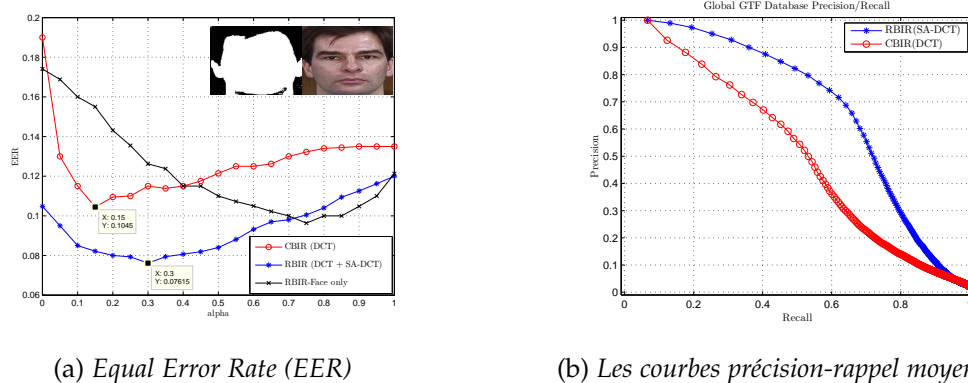
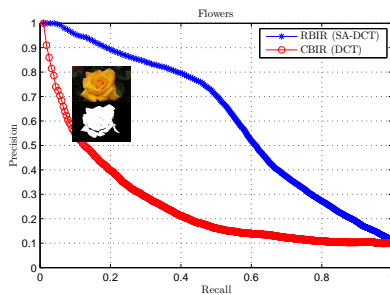


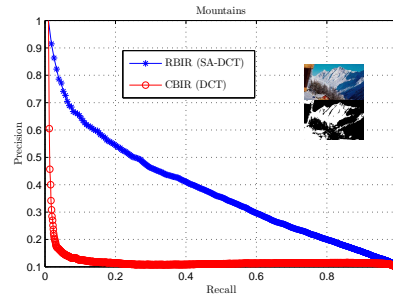
Figure 3.18 – Comparaison des résultats pour la base de données GTF : a) EER b) Les courbes précision-rappel moyennes.

3.4.4.3 Optimisation

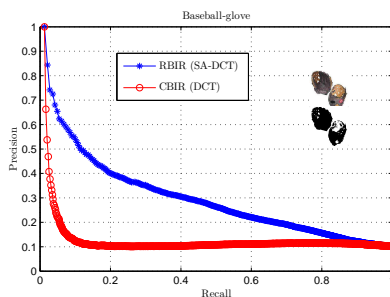
Dans ce qui suit on propose les courbes après une optimisation par la séparation de l'histogramme de l'objet, (premier-plan et arrière-plan), en histogramme issu des patterns (AC et DC) des blocs intérieurs et en histogramme issu des patterns (AC et DC) des segments des blocs frontières en employant en plus des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$, d'autres paramètres $\{ \gamma_{CF}, \gamma_{PF}, \gamma_{CB}, \gamma_{PB} \}$ voir (Sec.3.4.3). Les courbes des figures (Fig.3.19) et (Fig.3.20) sont pour les bases de données Corel-1000 et Caltech-256, alors que les courbes des figures (Fig.3.21) et (Fig.3.22) sont pour la base de données GTF.



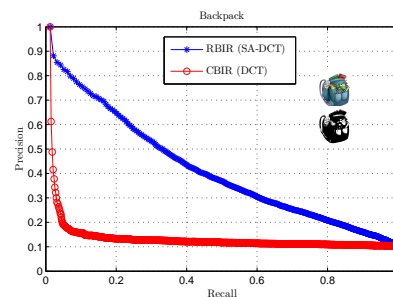
(a) P-R utilisant "Fleurs" de Corel-1000



(b) P-R utilisant "Montagnes" de Corel-1000

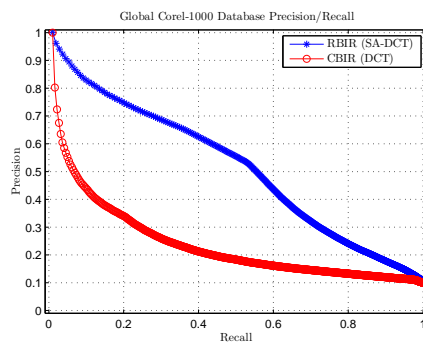


(c) P-R utilisant "Gants-Baseball" de Caltech-256

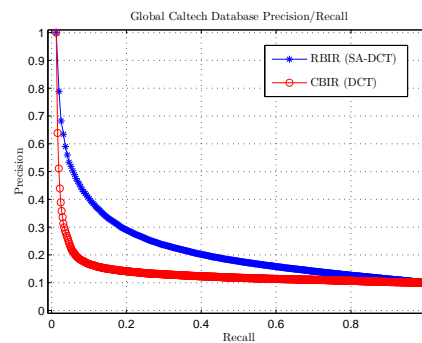


(d) P-R utilisant "Sacs-à-dos" de Caltech-256

Figure 3.19 – Optimisation : Les courbes précision-rappel moyennes entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement). Les catégories d'images (a)"Fleurs" et (b)"Montagnes" sont de la base de données Corel-1000, et les catégories d'images (c)"Gants de baseball" et (d)"Sacs-à-dos" sont de la base de données Caltech-256



(a) P-R de la base de données Corel-1000



(b) P-R de la base de données Caltech-256

Figure 3.20 – Optimisation : Les courbes précision-rappel moyennes : (a)Base de données Corel-1000, (b) Base de données Caltech-256, entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement)[86]

D'après les résultats trouvés, on remarque que les paramètres $\{ \gamma_{CF}, \gamma_{PF}, \gamma_{CB}, \gamma_{PB} \}$ ont pu améliorer les résultats des courbes précision/rappel pour les trois bases de données Corel-1000, Caltech-256 et GTF d'une manière significative.

Par conséquent, on constate que notre système offre plus de souplesse par l'emploi des paramètres qui permettent d'augmenter la performance de la recherche.

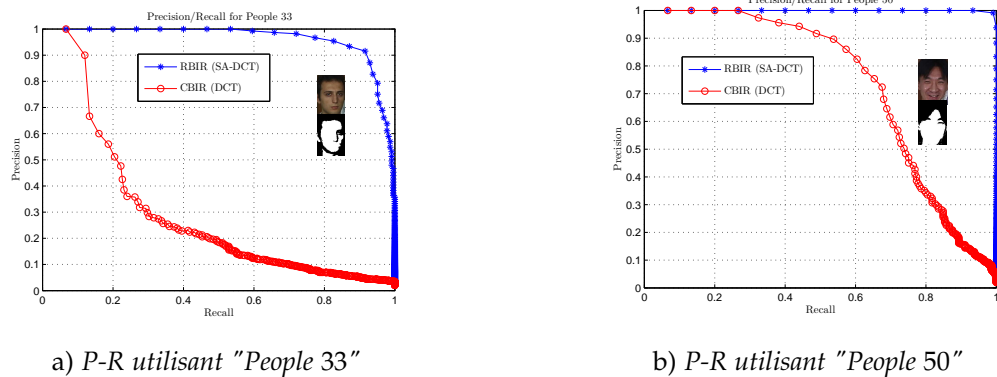
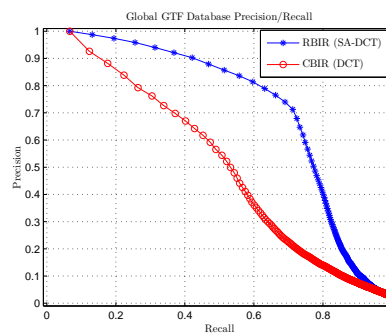


Figure 3.21 – Optimisation : Les courbes précision-rappel moyennes entre notre approche basée-région (DCT+SA-DCT) et l'approche classique (CBIR avec DCT seulement). Les catégories d'images (a)"People 33" (b)"People 50" sont de la base de données GTF



(b) Les courbes précision-rappel moyennes

Figure 3.22 – Optimisation : Comparaison des résultats pour la base de données GTF : Les courbes précision-rappel moyennes.

3.4.4.4 Comparaison avec des méthodes RBIR déjà existantes

Après avoir confirmé que le *Système avec paramètres globaux* proposé (DCT+SA-DCT), est plus performant que le système classique (CBIR avec DCT seulement), on a comparé celui optimisé avec six méthodes RBIR déjà existantes : Significant Region-Based Image Retrieval (SRBIR) [60], Adaptive Region Matching (ARM) [82], MN-IRM [83], SIMPLIcity [77], Robust Local Binary Pattern (RLBP) [63] et MN-MIN [25]. Les détails spécifiques peuvent être trouvés en références liées.

D'après les tableaux (Tab.3.8), et (Tab.3.9), on remarque qu'en général le système RBIR proposé fonctionne mieux que les autres systèmes RBIR existants, sauf pour cinq classes de la base de données Corel-1000, ceci est dû à la mauvaise segmentation des images de ces classes. Pour la base de données Caltech-256, notre système RBIR (DCT+SA-DCT) surpasse dans l'ensemble les autres méthodes (RBIR et CBIR).

La figure (Fig.3.23) représente une comparaison schématique du MAP pour chaque catégorie entre la méthode proposée (DCT+SA-DCT) et quatre méthodes dans le domaine

ID Catégorie	1	2	3	4	5	6	7	8	9	10
SA-DCT	75.45	52.50	46.30	97.25	100	90.20	90.50	88.25	61.45	76.95
MN-ARM [82]	74.23	46.35	74.51	80.60	99.80	62.42	94.84	92.91	61.80	78.88
MN-IRM [83]	73.80	42.86	74.32	74.24	99.80	61.00	95.23	93.49	51.33	66.54
SIMPLIcity [77]	52.73	25.52	56.54	49.40	98.50	45.51	78.73	78.28	39.00	39.13
MN-MIN [25]	43.40	29.43	31.82	46.63	90.11	33.07	64.94	63.62	27.90	35.21
DCT [86]	52.45	27.20	27.10	36.95	93.05	46.90	53.45	61.65	17.10	37.45

Table 3.8 – Précision Moyenne(AP) des différentes méthodes pour la base de données Corel-1000

ID Catégorie	1	2	3	4	5	6	7	8	9	10
SA-DCT	35.0	32.89	71.48	34.37	52.65	33.43	44.76	49.92	44.06	29.60
MN-ARM [82]	22.81	17.63	38.41	40.83	36.28	18.37	26.51	46.98	18.95	15.41
MN-IRM [83]	21.53	16.96	35.73	40.39	31.52	16.00	24.20	44.68	18.94	14.94
SIMPLIcity [77]	21.51	16.88	34.95	40.33	31.33	17.87	24.06	44.69	18.86	14.76
MN-MIN [25]	21.30	16.28	33.45	40.38	33.17	18.09	24.00	44.57	18.90	18.80
DCT [86]	25.31	18.04	22.03	31.64	23.43	17.42	26.56	30.54	26.01	18.12

Table 3.9 – Précision Moyenne(AP) des différentes méthodes pour la base de données Caltech-256

RBIR (SRBIR, ARM, SIMPLIcity, RLBP) et une méthode dans le domaine CBIR (CBIR avec DCT seulement), pour les bases de données Corel-1000 et Caltech-256.

On peut conclure d'après les résultats expérimentaux, que le système proposé améliore la

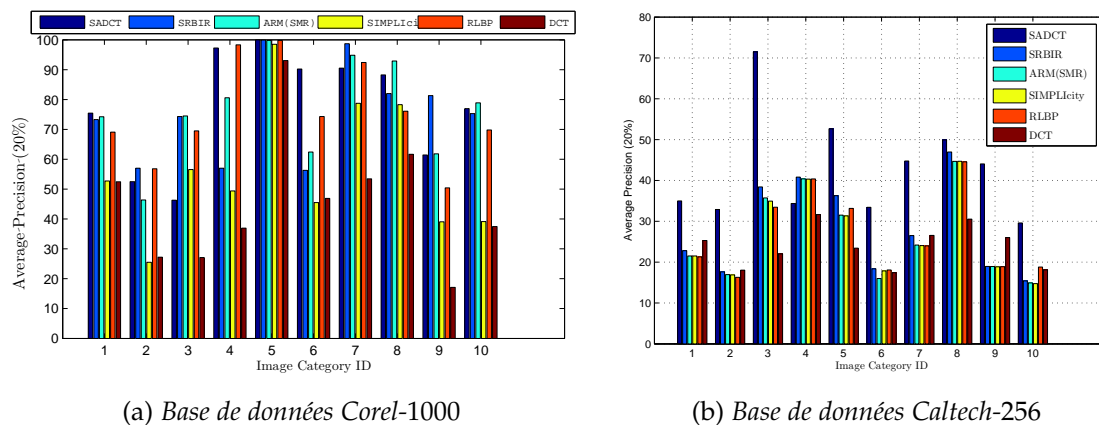


Figure 3.23 – Mean Average Precision (MAP) (a) Base de données Corel-1000, (b) Base de données Caltech-256

performance sur les bases de données COREL-1000 et Caltech-256. En outre, on a utilisé un nombre minimal de coefficients AC et un nombre faible de coefficients d'histogramme pour réduire le temps de calcul.

3.4.4.5 Évaluation de la robustesse

Dans cette section, nous analysons la robustesse du *Système avec paramètres globaux optimisé* RBIR basé sur la transformée SA-DCT. La figure (Fig.3.24) et la figure (Fig.3.25) illustrent les résultats avec les bases de données Corel-1000 et Caltech-256. Ces sous figures dans la première et la troisième colonne montrent le changement du rang de l'image souhaitée dans la recherche d'images, et dans la deuxième et la quatrième colonnes montrent la distance chi-square (éq.2.5) entre les images altérées et l'image souhaitée par l'incrément de la variation des images.

Il est montré que l'augmentation du pourcentage brighten/darken, de la taille de la fenêtre du filtre Gaussian et de la rotation d'angle pour certains degrés, que le rang de l'image souhaitée reste dans une position acceptable et que la distance entre l'image altérée et l'image désirée reste faible. Toutefois, notre système est robuste avec les différentes variations, la variation du flou et de la rotation.

Les résultats dans la figure (Fig.3.26) démontrent quelques exemples de requêtes sur la base de données Corel-1000.

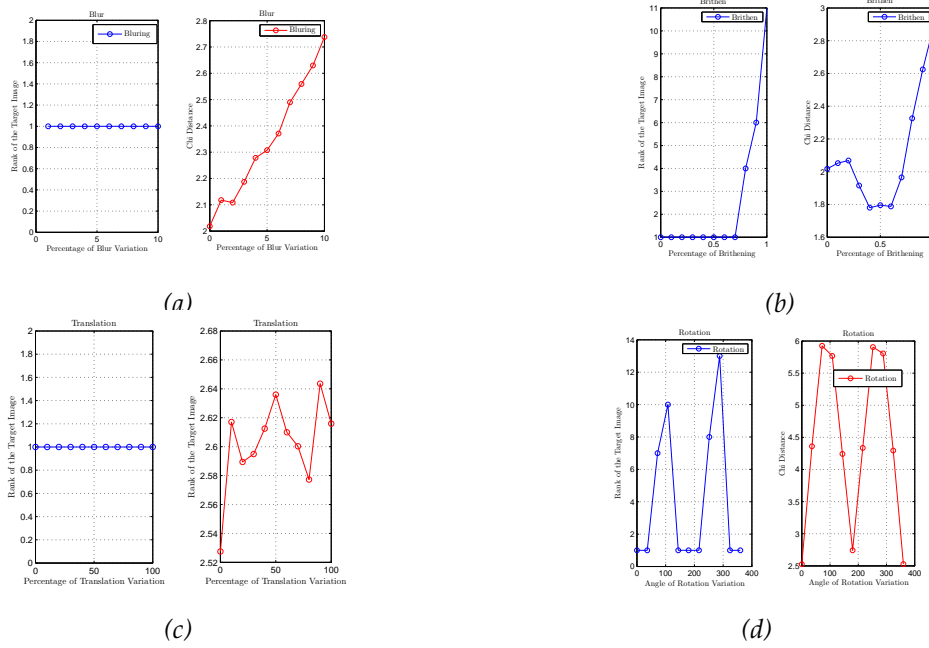


Figure 3.24 – Robustesse de notre Système avec paramètres globaux pour les variations d’images pour la base de données Corel-1000

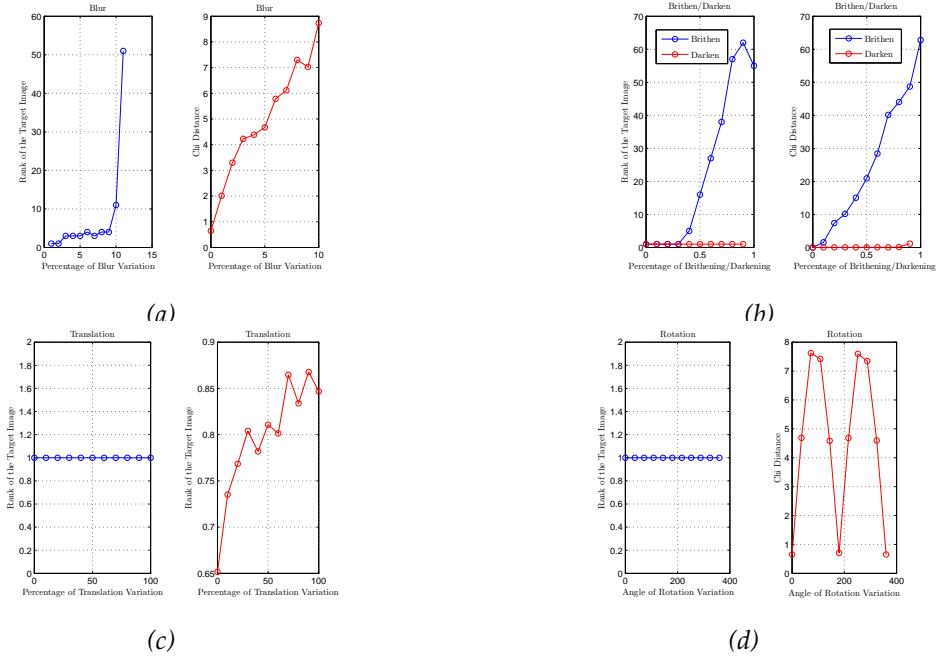


Figure 3.25 – Robustesse du Système avec paramètres globaux pour les variations d’images pour la base de données Caltech-256

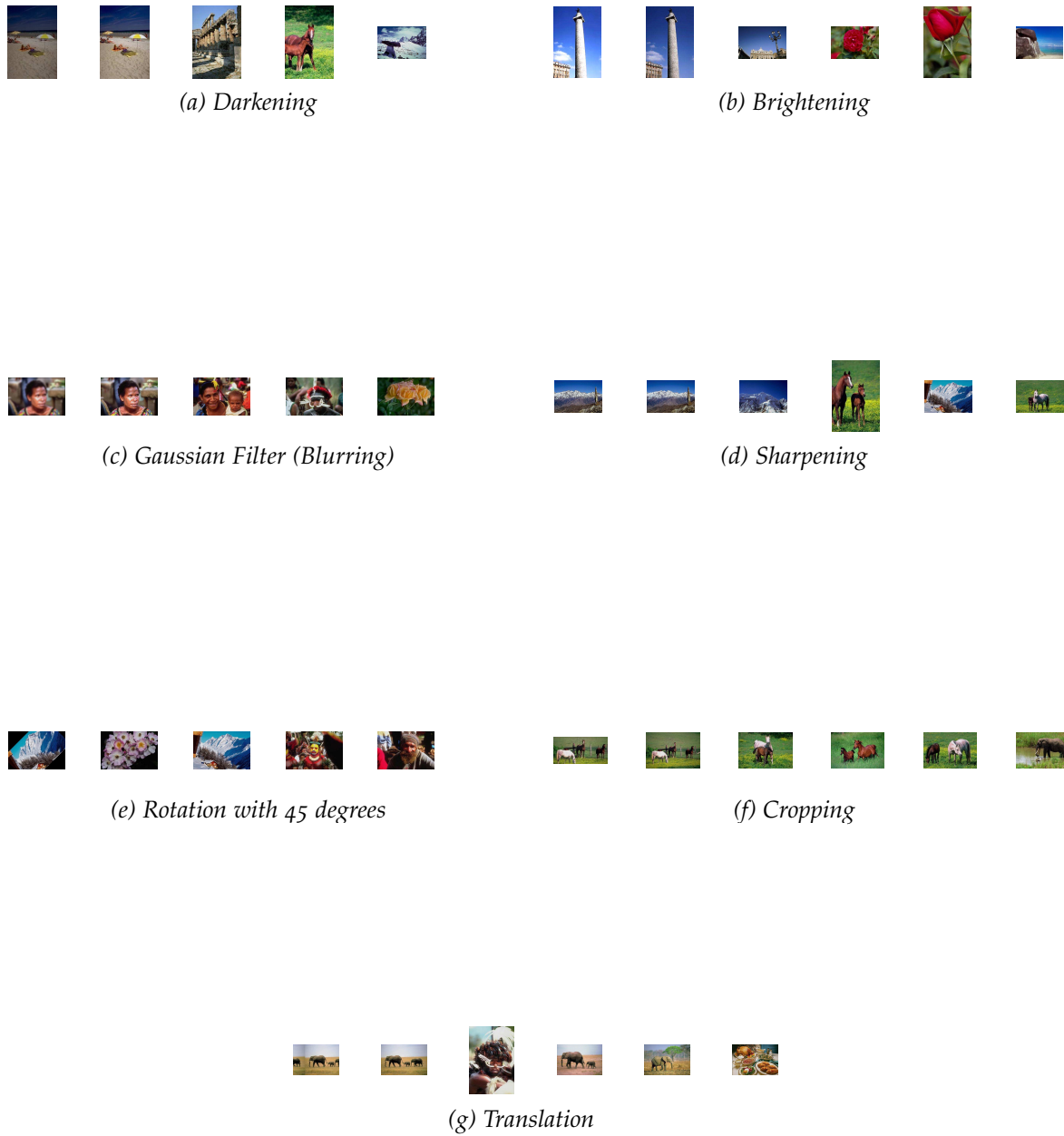


Figure 3.26 – Quelques exemples d'évaluation de robustesse sur la base de données Corel-1000. La première image est l'image requête, ensuite les 5 premières images recherchées par l'utilisation du Système avec paramètres globaux proposé.

3.4.4.6 Conclusion partielle

Dans cette partie une comparaison entre le système proposé et le système classique a été faite. D'après les résultats obtenus, on remarque que le *Système avec paramètres globaux* a pu contourner les problèmes posés par le système classique DCT (CBIR avec DCT seulement). Ceci est dû à la souplesse du système proposé (DCT+SADCT), qui offre une possibilité de variation de plusieurs paramètres comparativement avec le système classique (CBIR avec DCT seulement) qui offre la possibilité de gérer un seul paramètre. En plus, l'emploi de la SA-DCT permet la séparation des objets de l'image et le traitement de cette dernière d'une manière sémantique.

Le système proposé est comparable avec des méthodes connues dans le domaine de l'indexation et de la recherche d'images par région, comme la méthode SIMPLicity [77].

3.4.5 Optimisation de la mesure de similarité

Dans cette section, on présente une expérimentation qui permet de réduire le temps de recherche tout en garantissant les mêmes performances du système proposé. Elle consiste à limiter le calcul de similarité seulement entre les régions similaires. C'est-à-dire, la comparaison entre la région premier-plan de l'image requête se fait seulement avec les régions premiers-plan des images de la base de données. Dans le même ordre d'idées, la comparaison entre la région arrière-plan de l'image requête se fait seulement avec les régions arrière-plan des images de la base de données. Ceci permet d'éviter le calcul des distances entre les régions qui ne correspondent pas et donc gagner en temps de recherche. La figure (Fig.3.27) montre la manière dont le calcul de similarité est fait pour économiser le temps de recherche toute en gardant une qualité acceptable du système proposé.

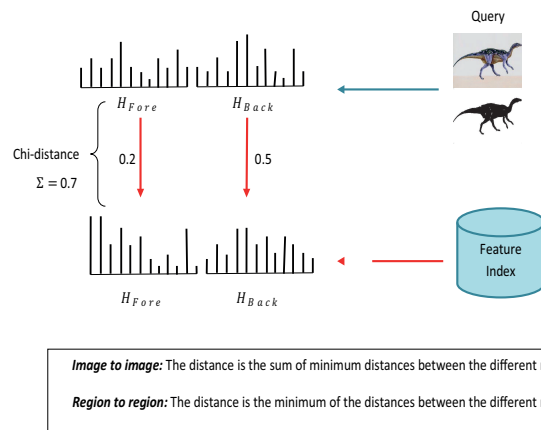


Figure 3.27 – Optimisation de la mesure de similarité : exemple de calcul de similarité entre les images.

Pour confirmer les propositions dites en-haut, la figure (Fig.3.28) montre une comparaison entre la courbe précision-rappel pour la base de données Corel-1000 en considération de la mesure de similarité *image par image* qui prend en compte la similarité entre les régions similaires et non-similaires, et la courbe précision-rappel en considération de la mesure de similarité *image par image* qui prend en compte la similarité entre seulement les régions similaires.

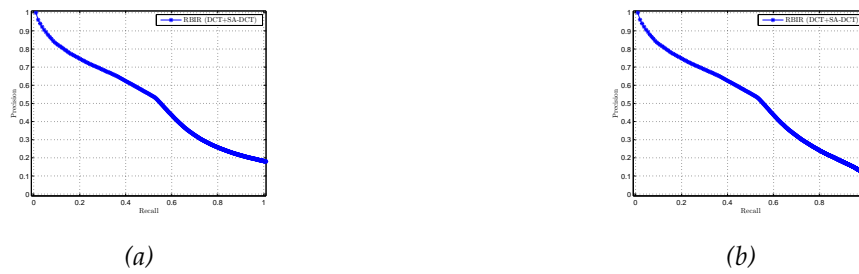


Figure 3.28 – Les courbes précision-rappel pour la base de données Corel-1000 calculées par la mesure de similarité image par image. (a) Chaque région de l’image requête est comparée avec toutes les régions de la base de données (similaires et non-similaires). (b) Chaque région de l’image requête est comparée avec seulement les régions similaires de la base de données.

D’après la figure (Fig.3.28), on remarque que la mesure de similarité entre toutes les régions, (similaires et non-similaires), n’apporte pas grand-chose en matière de précision-rappel et de qualité de la recherche. En plus, elle est nuisible en terme de temps de recherche, car elle prend presque le double du temps que prend la mesure de similarité entre seulement les régions similaires, pour aboutir à la même performance.

3.4.6 Type de recherche

Comme il a été dit précédemment, le système RBIR (SADCT+DCT) proposé permet une mesure de similarité *région par région* ou bien *image par image* selon les caractéristiques statistiques de l'image requête. Ceci permet de réduire l'influence négative des régions en interférences et la perte des informations importantes. Dans ce qui suit, on va évaluer le système avec paramètres globaux optimisé en considérant les aspects précédents.

3.4.6.1 Réduction de l'influence négative des régions en interférences

Puisque la mesure de similarité *image par image* incorpore les propriétés des différentes régions, ceci implique des régions en interférences. Toutefois, la mesure de similarité *région par région* peut résoudre ce problème.

Considérant les catégories Plage et Bâtiment de la base de données Corel-1000 qui ont des régions en interférences. Les figures (3.29(a) et 3.29(b)) donnent une comparaison entre les résultats de notre système avec paramètres globaux optimisé selon que la recherche est *région par région* ou bien *image-par-image* sur les catégories Plage et Bâtiment respectivement. On dénote les résultats de comparaisons par : Beach Region-to-Region, Beach Image-to-Image, Building Region-to-Region, Building Image-to-Image. La comparaison sur la figure (Fig.3.29) montre que par la recherche des images les plus similaires $P(N)$ ($N = 10 : 10 : 100$) la mesure de similarité *région par région* a une supériorité consistante par rapport à *image par image*.

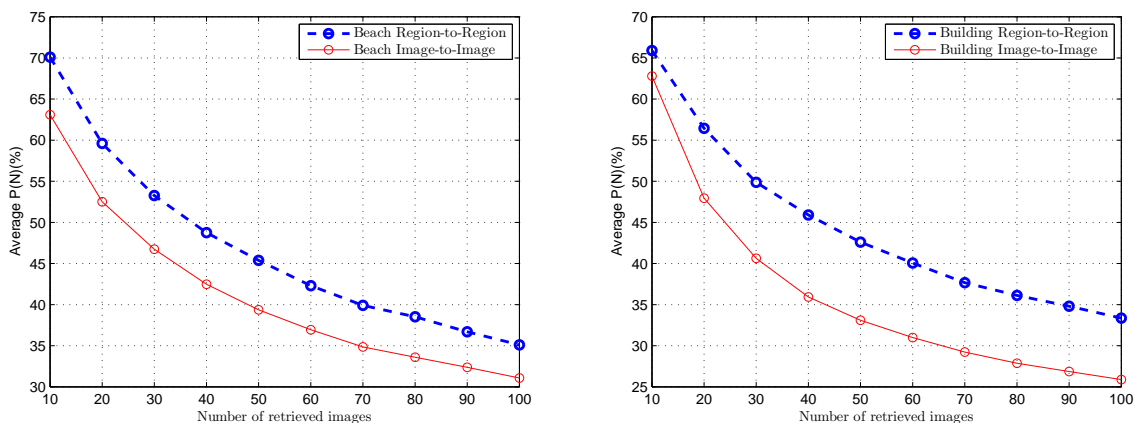


Figure 3.29 – La moyenne $P(N)$ ($N = 10 : 10 : 100$) (%) de la recherche *région-par-région* et *image-par-image*. (a) Plage, (b) Bâtiment, de la base de donnée Corel-1000

Pour démontrer que la mesure de similarité *région par région* peut effectivement diminuer l'influence négative des régions en interférences, on donne un exemple pour vérifier ce fait. La figure (Fig.3.30(a)) donne un exemple de requête en utilisant la mesure de similarité *image par image*. La première image est la requête associée au sable de la plage, ciel bleu et les gens, sa sémantique dans la catégorie prédéfinie de la base de données

Corel-1000 est la Plage, les sept dernières images sont les images recherchées. Il y a quatre images non pertinentes qui ne sont pas du sable et du ciel bleu, mais des bâtiments et du ciel bleu. Par conséquent, le ciel bleu est la région la plus similaire pour les deux catégories. On appelle ce type de région par la zone d'interférence. La figure (Fig.3.30(b)) montre les résultats de recherche *région par région*, qui démontre que par la sélection de l'objet, la performance du système a été considérablement augmentée.

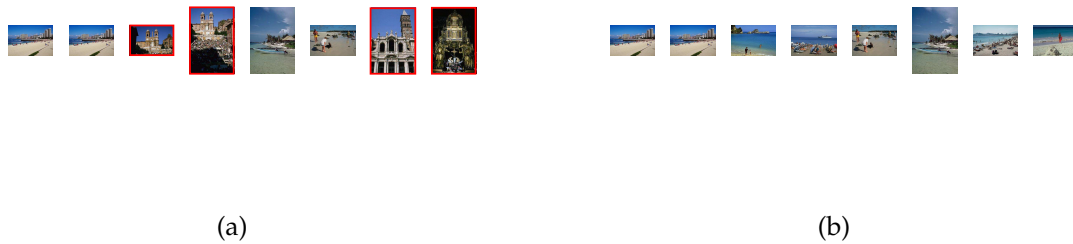


Figure 3.30 – Réduction de l'influence négative des régions en interférences. (a) La première image est l'image requête, les sept autres images sont le résultat RBIR du système proposé basé sur la mesure de similarité image par image, (b) La première image est l'image requête, les sept autres images sont le résultat RBIR du système proposé basé sur la mesure de similarité région par région

3.4.6.2 Diminution de l'influence négative de la perte importante d'information

Pour les images sans région principale, si on adopte la mesure de similarité *région par région* pour rechercher des images pertinentes, cela se traduira par des résultats insatisfaisants de recherche en raison de la perte importante d'information. Considérant les catégories Africains et Plats de la base de données Corel-1000 qui sont des images sans objet principal, on évaluera notre système sur les deux catégories. Dans le même ordre d'idées, les figures (3.31(a) et 3.31(b)) donnent une comparaison entre les résultats de notre système avec paramètres globaux optimisé selon que la recherche est *région par région* ou bien *image-par-image* sur les catégories Africains et Plats respectivement. On dénote les résultats de comparaisons par : African Region-to-Region, African Image-to-Image, Food Region-to-Region, Food Image-to-Image. La comparaison sur la figure (Fig.3.31) montre que par la recherche des images les plus similaires P(N) (N = 10 : 10 :100), la mesure de similarité *image par image* a une supériorité considérable par rapport à *région par région*.

Pour vérifier davantage que la mesure de similarité *image par image* peut effectivement diminuer l'influence négative de la perte importante d'information, on donne un exemple pour démontrer ce fait. La figure (Fig.3.32(a)) montre les résultats de recherche de l'image Africain par la mesure de similarité *région par région*. Il est évident que les résultats ne sont pas satisfaisants et il y a deux images non pertinentes pour les sept premières images recherchées. La figure (Fig.3.32(b)) indique que la mesure de similarité *image par image*

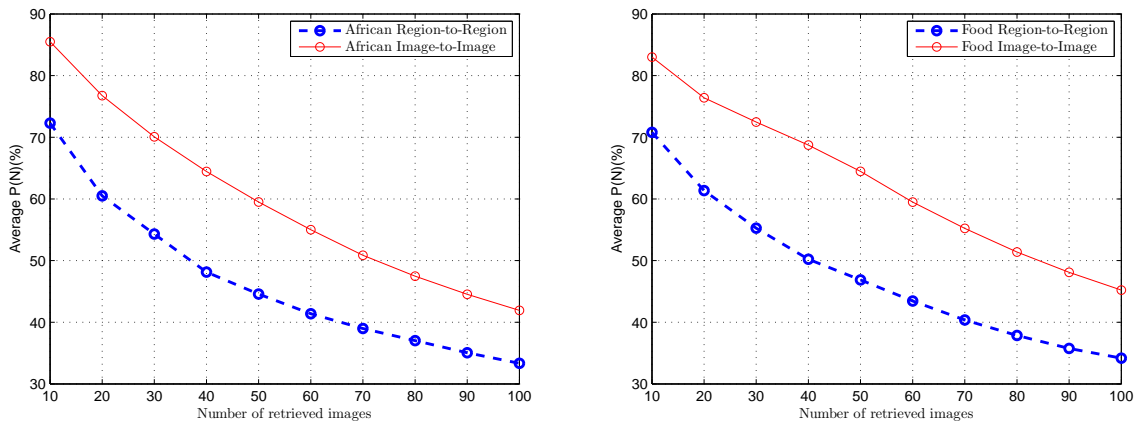


Figure 3.31 – La moyenne $P(N)$ ($N = 10 : 10 : 100$) (%) de la recherche région-par-région et image-par-image. (a) Africain, (b) Plat, de la base de donnée Corel-1000

améliore la recherche.

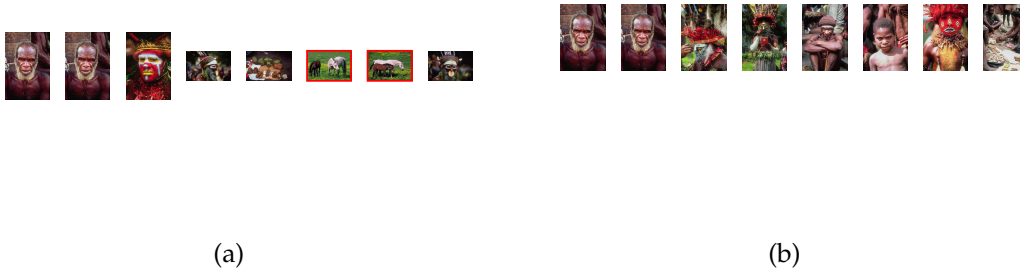


Figure 3.32 – Diminution de l'influence négative de la perte importante d'information. (a) La première image est l'image requête, les sept autres images sont le résultat RBIR du système proposé basé sur la mesure de similarité région par région, (b) La première image est l'image requête, les sept autres images sont le résultat RBIR du système proposé basé sur la mesure de similarité image par image

De la figure (Fig.3.32(b)), on peut déduire que pour les images sans région principale, la mesure de similarité *image par image* est plus raisonnable.

3.5 CONCLUSION

Dans ce chapitre, une méthode d'indexation d'images par région dans le domaine compressé est proposée.

Tout d'abord, on a introduit le concept du RBIR. On a parlé premièrement, de quelques travaux connexes des systèmes d'indexation et de recherche d'images par région. Ensuite, on a détaillé le principe de la méthode proposée basée sur la SA-DCT. Enfin, on a évoqué les résultats expérimentaux.

Deux systèmes sont extraits de la méthode proposée, *Système avec paramètres globaux* et *Système avec paramètres adaptés pour chaque classe* (voir Annexe B) selon que les paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ utilisés sont fixes ou variables pour chaque classe. Une méthode d'optimisation a été proposée pour améliorer la performance de la recherche. Elle consiste à séparer l'histogramme de l'objet, (premier-plan et arrière-plan), en histogramme issu des patterns (AC et DC) des blocs intérieurs et en histogramme issu des patterns (AC et DC) des segments des blocs frontières en employant en plus des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$, d'autres paramètres $\{ \gamma_{CF}, \gamma_{PF}, \gamma_{CB}, \gamma_{PB} \}$. Le système proposé a pu corriger les défauts du système classique (CBIR avec DCT seulement).

Une méthode d'optimisation dans le calcul de la mesure de similarité a été adopté par notre système. Elle consiste à comparer seulement entre les régions similaires des images, ceci permet de réduire le temps de recherche tout en conservant la performance du système. Les résultats expérimentaux montrent que la comparaison entre toutes les régions, (similaires et non-similaires), est sans influence significative sur la courbe précision-rappel. En plus, elle est gourmande en temps de recherche.

Deux types de mesure de similarité sont adoptées par notre système RBIR, *image par image* et *région par région*. Les résultats expérimentaux montrent que ces deux mesures permettent de réduire l'influence négative des régions en interférences et diminuent l'influence nocive de la perte importante d'information.

CONCLUSION

4

Basé sur une méthode classique d'indexation d'image par le contenu (CBIR avec DCT seulement), un système d'indexation d'image par région (RBIR avec DCT et SA-DCT) est proposé dans le présent document. Nous avons proposé une nouvelle vision, système d'indexation d'image basé-région, qui emploie la DCT et le Shape-Adaptive DCT. Par conséquent, le système peut faire une recherche d'une région sans avoir référence à des informations sur d'autres régions de l'image.

Le chapitre 2 de cette partie nous a introduits au système d'indexation d'image par le contenu basé sur la DCT. On a parlé brièvement des systèmes d'indexation d'images par le contenu, puis on a introduit la manière pour calculer le descripteur d'images dans le domaine DCT, l'histogramme des ACs et l'histogramme des DCs et leurs combinaisons pour construire le descripteur global. On a introduit aussi les mesures de similarité, ceci permet une comparaison des informations statistiques des images dans un chemin quantitatif par le calcul de la distance entre les histogrammes en utilisant une norme acceptable. À la fin de ce chapitre on a détaillé la manière pour évaluer la performance des systèmes d'indexation, précision-rappel, MAP, ARR et EER.

Au chapitre 3, on a détaillé l'aspect théorique de la méthode proposée. On s'est d'abord familiarisé aux principes de base d'indexation d'image par région. On a parlé du principe de l'approche proposée en utilisant la SA-DCT. On a ensuite évoqué brièvement la méthode utilisée pour faire la segmentation et on a donné la référence du code utilisé. Par la suite, une méthode de pré-traitement qui consiste à faire une normalisation puis une quantification des coefficients DCT et SA-DCT a été bien détaillée.

À la fin de ce chapitre, on a présenté les résultats expérimentaux. Deux systèmes sont extraits de la méthode proposée, *Système avec paramètres globaux* et *Système avec paramètres adaptés pour chaque classe* (Annexe B) selon que les paramètres utilisés sont globaux ou adaptés pour chaque classe. Les résultats présentés démontraient une amélioration au niveau de la précision-rappel, comparativement par la méthode d'indexation conventionnelle par le contenu (CBIR avec DCT seulement). En effets, l'indexation basée-objet dans le domaine transformé (*Système avec paramètres adaptés pour chaque classe*) voir (Annexe B), a contribué à une recherche dans certaines classes d'environ 100%.

Une optimisation dans le calcul de la mesure de similarité entre les images consiste à comparer entre seulement les régions similaires. Les résultats expérimentaux montrent

un gain significatif en temps de recherche tout en gardant les mêmes performances du système. En plus, deux mesures de similarité sont adoptées par le système proposé *image par image* et *région par région* selon que l'utilisateur sélectionne une région (arrière-plan ou premier-plan) de l'image ou non. Les résultats montrent que ces deux mesures permettent de réduire l'influence négative des régions en interférences et diminuent l'influence de la perte importante d'information.

Les résultats expérimentaux indiquent la justesse et l'efficacité de l'approche proposée, un taux de recherche élevé et moins de complexité. En plus, ils sont comparables à des méthodes de référence dans le domaine [77].

Dans le futur, on va s'intéresser aux images qui ont plus qu'un objet principal, en intégrant les propriétés de toutes les régions de l'image par l'insertion de l'indice d'importance de chaque région [82] (RII, Region Importance Index). En le comparant avec IRM (Integrated Region Matching) [77] qui est basé sur le pourcentage de la région, le RII prend en compte le pourcentage de la région ainsi que sa position. Ceci réduit les effets indésirables de la segmentation imprécise.

On s'intéressera aussi au problème du retour de pertinence (relevante feedback) et comment il peut aider à la prise en charge des besoins des utilisateurs. Contrairement aux méthodes classiques qui ne considèrent que l'exemple, on introduira l'approche qui permet d'exploiter à la fois l'exemple et le contre-exemple pour formuler et raffiner les requêtes [50].

Des efforts supplémentaires seront investis pour appliquer la méthode proposée sur des séquences vidéo.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Dans cette thèse, on a introduit, analysé et étudié la compression des vidéos et l'indexation des images basées-régions dans le domaine transformé. Ce travail se concentre principalement sur les algorithmes basés sur la DCT et le Shape-Adaptive DCT (SA-DCT), largement utilisé dans la compression des images et des vidéos. La partie (I) introduit un nouveau système de compression basé-région des séquences d'images monoculaires et stéréoscopiques dans le domaine transformé. Tandis que la partie (II) introduit un nouveau système d'indexation et de recherche des images basé-région et basé aussi sur la transformation SA-DCT. Les résultats expérimentaux des deux systèmes montrent la justesse et l'efficacité des approches proposées.

Un système conjoint entre les systèmes de compression et d'indexation proposés, consiste à extraire les descripteurs des images après exactement la transformation et la quantification. Les perspectives des futurs travaux pourront se concentrer sur les aspects suivants respectivement :

- Accélération du processus de codage proposé par l'utilisation des méthodes que celles de [91], [84].
- L'extension du système d'indexation et de la recherche des images aux séquences vidéo par l'utilisation des trames de référence (Intra-trames).
- L'intégration du système d'indexation et de la recherche de la vidéo dans le schéma de compression de la vidéo proposé après l'étape de transformation et de quantification.
- La compression du descripteur image/vidéo pour aboutir à un descripteur visuel compact (CDVS)[1, 2, 3].
- La comparaison entre la taille du descripteur codé et la taille de l'image compressée, car parfois il revient intéressant à envoyer un descripteur codé que d'envoyer une image compressée selon le type d'application au décodeur [21].

Glossaire et Annexes

GLOSSAIRE

- JPEG : Joint Photographic Expert Group.
- MPEG-4 : Motion Photographic Expert Group.
- DCT : Discret Cosinus Transform.
- SA-DCT : Shape Adaptive DCT.
- CPM : Circular Prediction Mapping.
- NCIM : Noncontractive Interframe Mapping.
- IFS : Iterative Fonction System.
- PSNR : Pic Signal Noise Ratio.
- QV : Quantification Vectorielle.
- RLE : Run Length Encoding.
- SA-spatial : Shape Adaptive Spatial.
- EM : Estimation Mouvement.
- CM : Compensation Mouvement.
- DCP : Disparity Compensated Prediction.
- MCP : Mouvement Compensated Prediction.
- QDE : Quadtree Disparity Estimation.
- QDC : Quadtree Disparity Compensation.
- CIF : Commun Intermediate Format.
- SIF : Source Input Format.
- BW : Band Width.
- Min-Part : Partitionnement quadtree de profondeur minimum.
- Max-Part : Partitionnement quadtree de profondeur maximum.
- CBIR : Content-Based Image Retrieval.
- RBIR : Région-Based Image Retrieval.
- MAP : Mean Average Precigion.
- P/R : Precision/Recall.
- EER : Error Equal Rate.
- FRR : False Rejected Rate.
- FAR : False Accepted Rate.
- ARR : Average Retrieval Rate.
- AC-pattern : Modèle du AC (Texture Pattern).
- DC-pattern : Modèle du DC (Color Pattern).
- H_{AC} : Histogramme des AC.
- H_{DC} : Histogramme des DC.
- H_{Full} : Histogramme Combiné global.
- H_{Fore} : Histogramme Combiné du premier-plan.
- H_{Back} : Histogramme Combiné de l'arrière-plan.
- H_{CF} : Histogramme Combiné issu des segments des blocs frontières du premier-plan.
- H_{PF} : Histogramme Combiné issu des blocs intérieurs du premier-plan.
- H_{CB} : Histogramme Combiné issu des segments des blocs frontières de arrière-plan.
- H_{PB} : Histogramme Combiné issu des blocs intérieurs de l'arrière-plan.
- POCS : Projections Onto Convex Sets.
- TV-Seg : Total Variation Segmentation Method.
- ARM : Adaptive Region Matching.

- RLBP :Robust Local Binary Pattern.
- SMR : Semantic Meaningful Region.
- SRBIR : Semantic Region-Based Image Retrieval
- IRM : Integrated Region Matching.
- WALRUS : WAveLet-based Retrieval of User-specified Scenes

ANNEXE A

ÉCHANTILLONS D'IMAGES DES DIFFÉRENTES BASES DE DONNÉES



Figure 1 – Échantillon d'images et leurs masques pour chaque classe de la base de données Corel-1000.

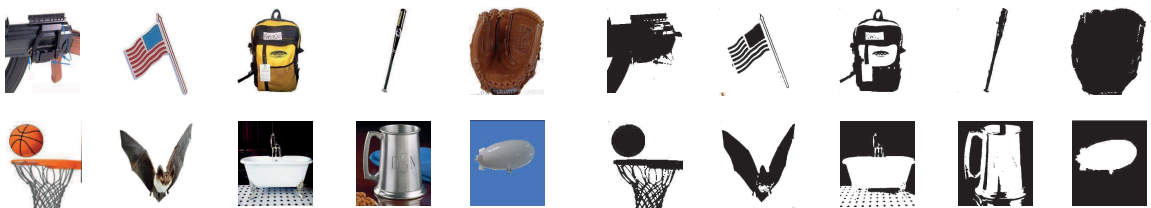


Figure 2 – Échantillon d'images et leurs masques pour chaque classe de la base de données Caltech-256.



Figure 3 – Échantillon d'images et leurs masques pour quelques classes de la base de données GTF.

ANNEXE B

SYSTÈME AVEC PARAMÈTRES ADAPTÉS POUR CHAQUE CLASSE (FIG.3.12)

Résultats avec les bases de données Corel-1000 et Caltech-256

La comparaison des résultats des courbes moyennes de précision-rappel, pour la base de données Corel-1000, entre l'approche proposée basée-objet et l'approche conventionnelle basée sur le contenu [86] est montrée dans la figure (Fig.2(a)). Les mêmes courbes, pour la base de données Caltech-256, sont montrées dans la figure (Fig.2(b)). La meilleure performance est obtenue par l'utilisation du descripteur global combiné (éq. 3.1) pour le premier-plan et l'arrière-plan ensemble.

Les paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ utilisés pour les expérimentations des deux bases de données Corel-1000 et Caltech-256 sont montrés dans les tableaux suivants (Tab.1), (Tab.2) respectivement.

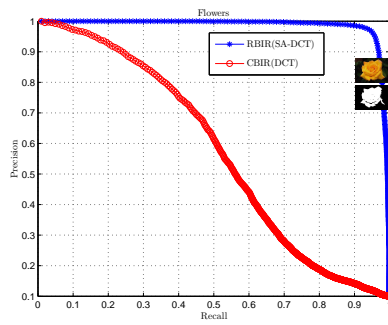
Classe	1	2	3	4	5	6	7	8	9	10
β	0.1	0.1	0.3	0.3	0.9	0.3	1	0.1	0.7	0.5
α_{Fore}	0.1	0.3	1	0.5	1	0.6	0.3	0.9	0.7	0.5
α_{Back}	0.1	0.3	0	1	1	1	0.4	0.9	0.7	0.1
$\alpha(DCT)$	0.7	0.1	0.3	0	0	0.6	0.4	0.2	0	0.1

Table 1 – $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ pour chaque classe pour la base de données Corel-1000

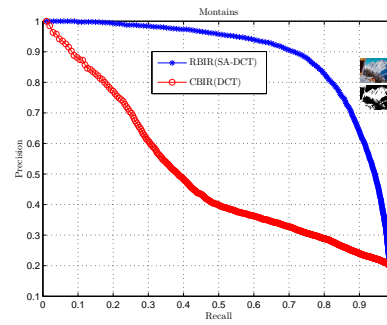
Classe	1	2	3	4	5	6	7	8	9	10
β	0	0.5	0.1	0.4	0.4	0.6	1	0.7	0.7	0.7
α_{Fore}	0	0	0.2	0.4	0.9	0.6	1	1	0.4	0.1
α_{Back}	0.7	0.8	0.9	0.9	0.9	0	0.4	0.9	1	0.1
$\alpha(DCT)$	0	0.2	0.4	0	0	0.6	0.1	0.6	0.7	0.1

Table 2 – $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ pour chaque classe pour la base de données Caltech-256

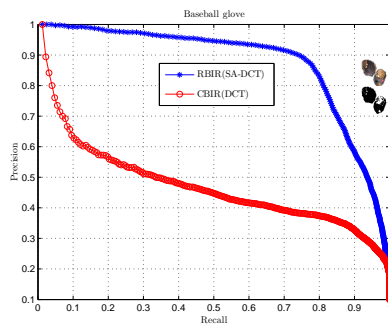
Comme il a été dit précédemment, les paramètres adaptés pour chaque classe permettent d'améliorer la qualité de la recherche d'une manière significative comparative-ment avec la figure (Fig.3.16). Il faut noter que pour la méthode classique DCT (CBIR avec DCT seulement), on a aussi utilisé un paramètre α , de l'équation (éq.2.2), adapté pour chaque classe.



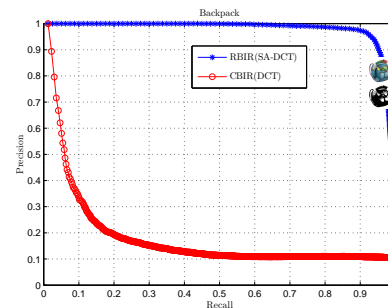
(a) P-R utilisant "Fleurs" de Corel-1000



(b) P-R utilisant "Montagnes" de Corel-1000

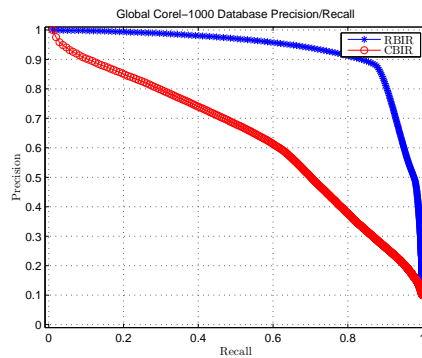


(c) P-R utilisant "Gants-Baseball" de Caltech-256

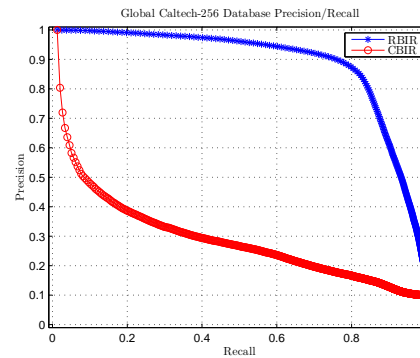


(d) P-R utilisant "Sacs-à-dos" de Caltech-256

Figure 1 – Les courbes précision-rappel moyennes entre l'approche basée-objet proposée (DCT+SA-DCT) et l'approche conventionnelle basée sur le contenu (CBIR avec DCT seulement). Les catégories d'images (a)"Fleurs" (b)"Montagnes" sont de la base de données Corel-1000, et les catégories d'images (c)"Gants de baseball" (d)"Sacs-à-dos" sont de la base de données Caltech-256



(a) P-R de la base de données Corel-1000

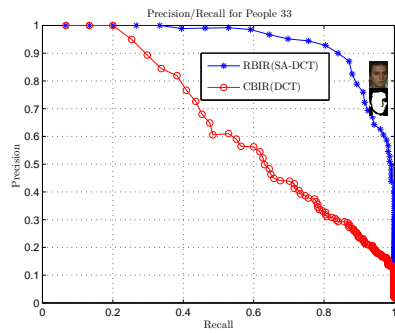


(b) P-R de la base de données Caltech-256

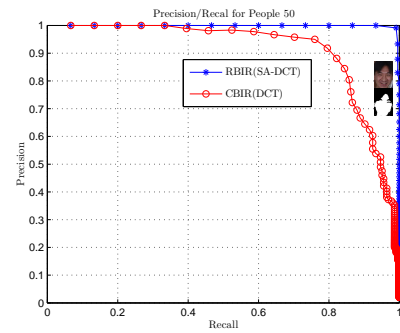
Figure 2 – Les courbes précision-rappel moyennes : (a)Base de données Corel-1000, (b) Base de données Caltech-256, entre l'approche proposée basée-objet (DCT+SA-DCT) et l'approche conventionnelle basée sur le contenu (CBIR avec DCT seulement)[86]

Résultats avec la base de données visage GTF

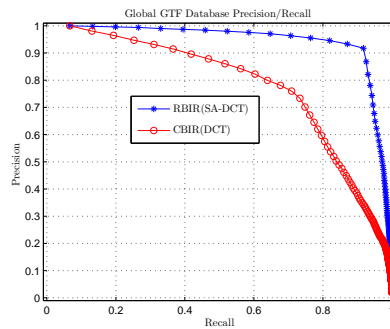
Pour la base de données visage GTF, la figure (Fig.3) illustre les courbes précision-rappel pour People 33, People 50 et la courbe précision-rappel moyenne de la base de données GTF en utilisant le système avec paramètres variables adaptés pour chaque classe. Une



a) P-R utilisant "People 33"



b) P-R utilisant "People 50"



c) Les courbes précision-rappel moyennes

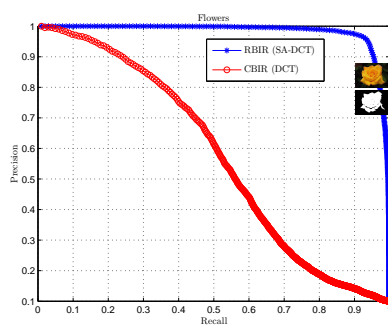
Figure 3 – Comparaison des résultats pour la base de données GTF : a) People 33 b) People 50 c) Les courbes précision-rappel moyennes.

étude comparative entre les figures (Fig.3.18, Fig.3.17) et la figure (Fig.3) montre que le *Système avec paramètres adaptés pour chaque classe* améliore les courbes précision-rappel pour presque la totalité des classes y compris People 33 et people 50. Par conséquent, une nette amélioration de la courbe précision-rappel moyenne pour la base de données GTF (Fig.3)(c).

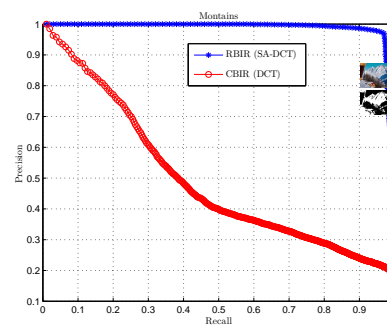
Optimisation

Dans ce qui suit on propose les courbes après une optimisation par la séparation de l'histogramme de l'objet, (premier-plan et arrière-plan), en histogramme issu des patterns (AC et DC) des blocs intérieurs et en histogramme issu des patterns (AC et DC) des segments des blocs frontières en employant en plus des paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$, d'autres paramètres $\{ \gamma_{CF}, \gamma_{PF}, \gamma_{CB}, \gamma_{PB} \}$ voir (Sec.3.4.3). Il faut noter que les paramètres $\{ \gamma_{CF}, \gamma_{PF}, \gamma_{CB}, \gamma_{PB} \}$ sont aussi adaptés pour chaque classe.

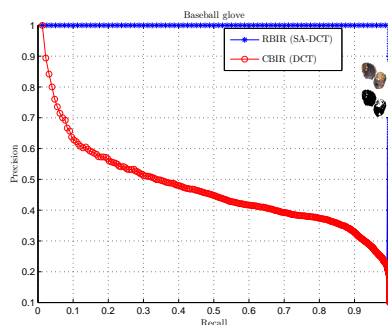
Les courbes des figures (Fig.4) et (Fig.5) sont pour les bases de données Corel-1000 et Caltech-256, alors que les courbes de la figure (Fig.6) sont pour la base de données GTF.



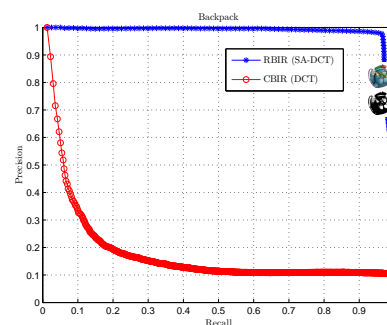
(a) P-R utilisant "Fleurs" de Corel-1000



(b) P-R utilisant "Montagnes" de Corel-1000



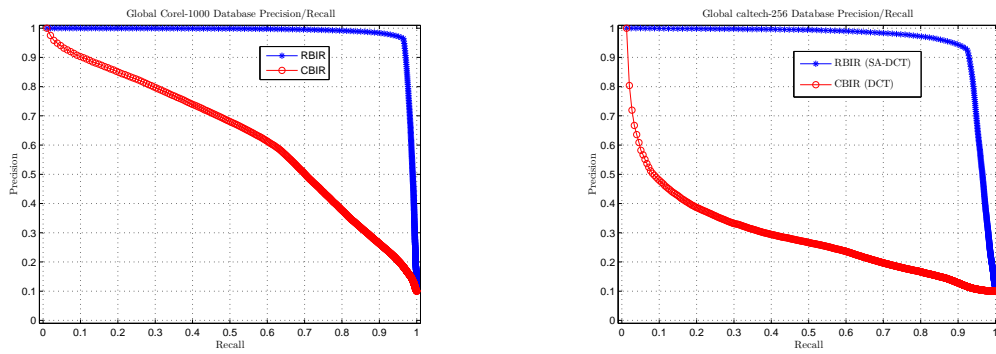
(c) P-R utilisant "Gants-Baseball" de Caltech-256



(d) P-R utilisant "Sacs-à-dos" de Caltech-256

Figure 4 – Optimisation : les courbes précision-rappel moyennes entre l'approche proposée basée-région (DCT+SA-DCT) et l'approche conventionnelle basée sur le contenu (CBIR avec DCT seulement). Les catégories d'images (a)"Fleurs" (b)"Montagnes" sont de la base de données Corel-1000, et les catégories d'images (c)"Gants de baseball" (d)"Sacs-à-dos" sont de la base de données Caltech-256

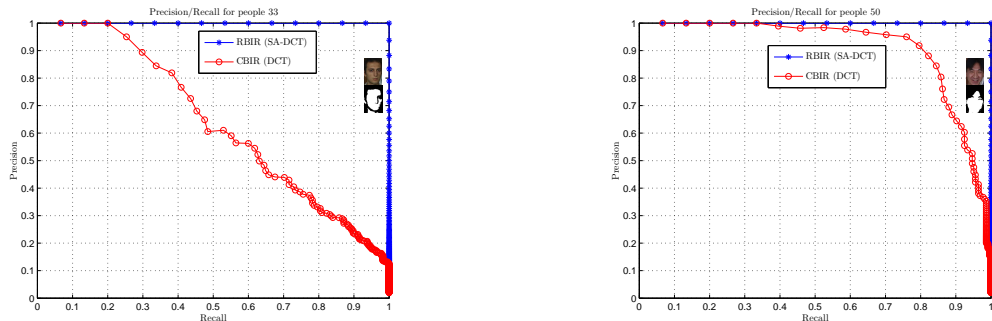
D'après les résultats illustrés dans les figures précédentes, on constate que les nouveaux paramètres $\{ \gamma_{CF}, \gamma_{PF}, \gamma_{CB}, \gamma_{PB} \}$ permettent d'augmenter les courbes précision-rappel pour le Système avec paramètres adaptés pour chaque classe d'une manière significative comparativement avec les figures (Fig.1,2,3). Les courbes de la méthode proposée (Fig.4(c)), (Fig.6(a)) et (Fig.6(b)) sont maintenant à 1, cela veut dire une recherche à 100%.



(a) P-R de la base de données Corel-1000

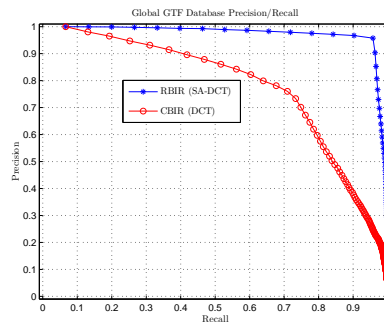
(b) P-R de la base de données Caltech-256

Figure 5 – Optimisation : les courbes précision-rappel moyennes : (a) Base de données Corel-1000, (b) Base de données Caltech-256, entre l'approche proposée basée-objet (DCT+SA-DCT) et l'approche conventionnelle basée sur le contenu (CBIR avec DCT seulement)[86]



a) P-R utilisant "People 33"

b) P-R utilisant "People 50"



c) Les courbes précision-rappel moyennes

Figure 6 – Optimisation : comparaison des résultats pour la base de données GTF : a) People 33 b) People 50 c) Les courbes précision-rappel moyennes.

Comparaison avec des méthodes RBIR déjà existantes

Passant maintenant à la comparaison du *Système avec paramètres adaptés pour chaque classe*, avec les six méthodes RBIR déjà existantes : Significant Region-Based Image Retrieval (SRBIR) [60], Adaptive Region Matching (ARM) [82], MN-IRM [83], SIMPLIcity [77], Robust Local Binary Pattern (RLBP) [63] et MN-MIN [25].

On remarque que notre méthode surpasse toutes les autres méthodes. En plus, si on

ID Category	1	2	3	4	5	6	7	8	9	10
SA-DCT	99.8	86.76	95.35	97.73	100	86.40	99.31	99.88	93.52	96.56
MN-ARM [82]	74.23	46.35	74.51	80.60	99.80	62.42	94.84	92.91	61.80	78.88
MN-IRM [83]	73.80	42.86	74.32	74.24	99.80	61.00	95.23	93.49	51.33	66.54
SIMPLicity [77]	52.73	25.52	56.54	49.40	98.50	45.51	78.73	78.28	39.00	39.13
MN-MIN [25]	43.40	29.43	31.82	46.63	90.11	33.07	64.94	63.62	27.90	35.21
DCT [86]	79.45	65.36	68.94	93.82	97.04	70.60	75.17	67.28	61.87	65.99

Table 3 – Précision Moyenne(AP) des différentes méthodes pour la base de données Corel-1000

ID Category	1	2	3	4	5	6	7	8	9	10
SA-DCT	99.07	95.85	98.93	80.40	93.0	98.07	90.60	95.60	93.64	96.47
MN-ARM [82]	22.81	17.63	38.41	40.83	36.28	18.37	26.51	46.98	18.95	15.41
MN-IRM [83]	21.53	16.96	35.73	40.39	31.52	16.00	24.20	44.68	18.94	14.94
SIMPLicity [77]	21.51	16.88	34.95	40.33	31.33	17.87	24.06	44.69	18.86	14.76
MN-MIN [25]	21.30	16.28	33.45	40.38	33.17	18.09	24.00	44.57	18.90	18.80
DCT [86]	46.7	28.43	29.17	34.07	53.9	29.79	50.08	45.49	28.5	56.26

Table 4 – Précision Moyenne(AP) des différentes méthodes pour la base de données Caltech-256

compare les résultats des tableaux (Tab.3) et (Tab.4) avec ceux des tableaux (Tab.3.8) et (Tab.3.9), on constate que le *Système avec paramètres adaptés pour chaque classe* a pu corriger les défauts du *Système avec paramètres globaux* d'une manière très attrayante.

La figure (Fig.7) illustre une comparaison du MAP pour chaque classe de la base de données Corel-1000 entre les résultats du *Système avec paramètres adaptés pour chaque classe* et d'autres méthodes RBIR déjà existantes (SRBIR, ARM, SIMPLicity, RLBP).

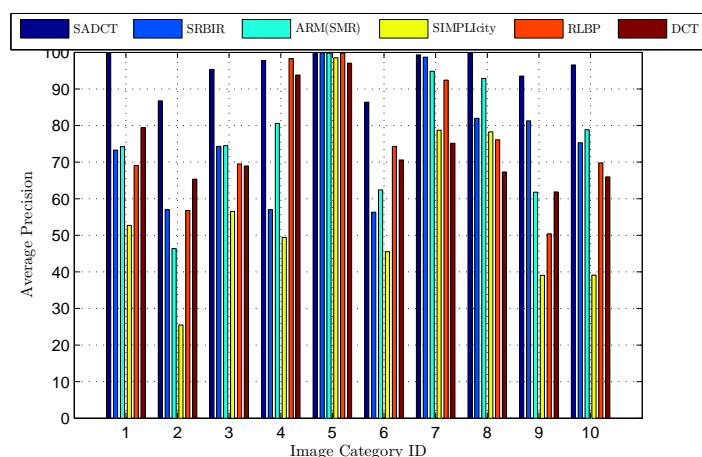


Figure 7 – Mean Average Precision (MAP) pour la Base de données Corel-1000

Évaluation de la robustesse

Dans cette section, nous analysons la robustesse du *Système avec paramètres adaptés pour chaque classe* RBIR basé sur la transformée SA-DCT. La figure (Fig.8) et la figure (Fig.9) illustrent les résultats avec les bases de données Corel-1000 et Caltech-256 respectivement. Ces sous figures dans la première et la troisième colonne montrent le changement du rang de l'image souhaitée dans la recherche d'images, et dans la deuxième et la quatrième colonnes montrent la distance chi-square (éq.2.5) entre les images altérées et l'image souhaitée par l'incrément de la variation des images.

Pour les deux bases de données, Corel-1000 et Caltech-256, on remarque que le *Système avec paramètres adaptés pour chaque classe* est aussi robuste devant les variations des images, et que ces variations n'affectent pas le rang de l'image souhaitée et que la distance entre l'image altérée et l'image désirée reste faible.

Les résultats dans la figure (Fig.10) démontrent quelques exemples de requêtes. La comparaison subjective entre les résultats des exemples de requêtes recherchées par le *Système avec paramètres globaux* et les résultats pour les mêmes types de requêtes recherchées par le *Système avec paramètres adaptés pour chaque classe*, démontre la robustesse du *Système avec paramètres adaptés pour chaque classe* en terme du rang de l'image recherchée, la distance entre l'image altérée et l'image souhaitée, et en plus la qualité de la recherche.

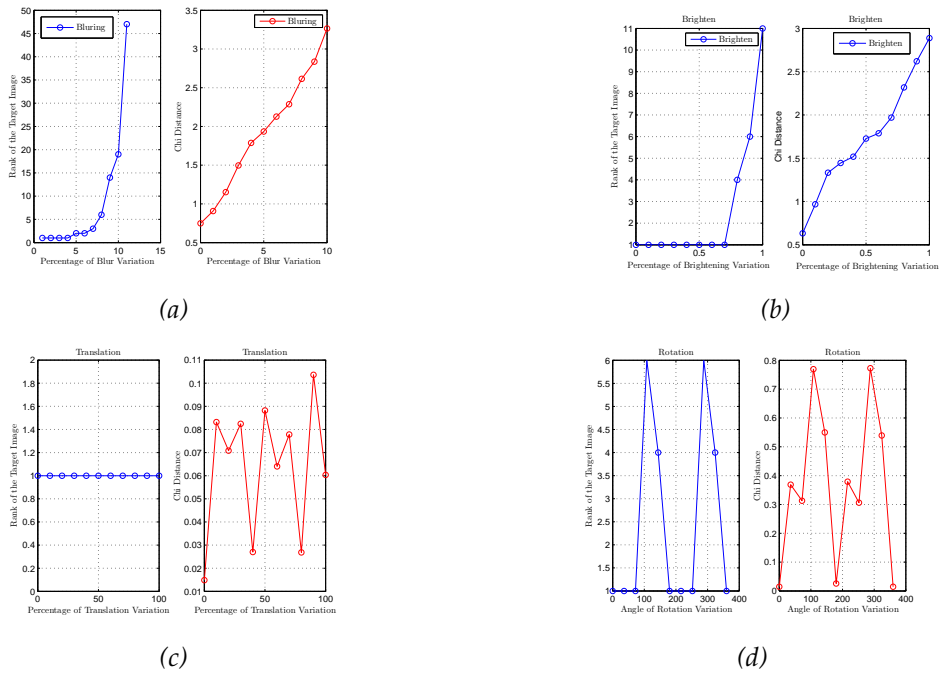


Figure 8 – Robustesse du Système avec paramètres adaptés pour chaque classe pour les variations d’images pour la base de données Corel-1000

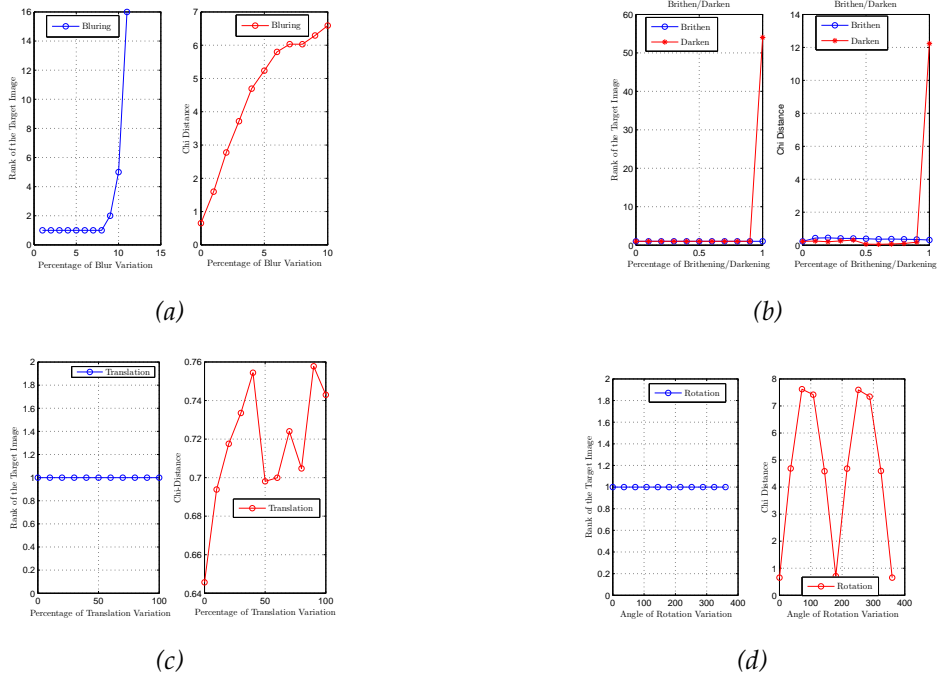


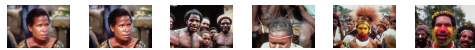
Figure 9 – Robustesse du Système avec paramètres adaptés pour chaque classe pour les variations d’images pour la base de données Caltech-256



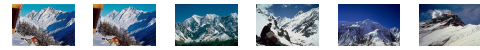
(a) Darkening



(b) Brightening



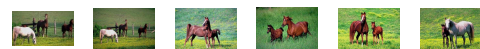
(c) Filtre Gaussian (Blurring)



(d) Sharpening



(e) Rotation avec 45 degrés



(f) Cropping



(g) Translation

Figure 10 – Quelques exemples d'évaluation de la robustesse sur la base de données Corel-1000. La première image est l'image requête, ensuite les 5 premières images recherchées par l'utilisation du Système avec paramètres adaptés pour chaque classe proposée.

Conclusion

Dans cette partie, on a introduit les paramètres nécessaires utilisés pour le Système avec paramètres adaptés pour chaque classe puis une comparaison entre ce système et le sys-

tème classique a été faite. Notons que pour le système classique DCT (CBIR avec DCT seulement), on a utilisé un paramètre α variable pour chaque classe, pour que la comparaison soit correctement faite. Les remarques dites auparavant (Sec.3.4.4) restent toujours valables.

On remarque aussi que le *Système avec paramètres adaptés pour chaque classe* a pu corriger les défauts du *Système avec paramètres globaux* pour les trois bases de données Corel-1000, Calthech-256 et GTF. Le problème qui se pose avec un tel système, est si on veut faire une recherche d'une image qui ne se trouve pas dans la base de données, alors il faut affecter cette image à une classe pour lui donner les paramètres $\{ \alpha_{Fore}, \alpha_{Back}, \beta \}$ appropriés. Pour un simple utilisateur du logiciel ceci n'est pas évident, car son souci est de faire seulement la recherche et non pas modifier dans les paramètres du logiciel.

Like.com [4], maintenant google shopping, ne permet pas aux utilisateurs de télécharger des images hors de la base de données existantes et donc l'utilisateur ne peut sélectionner que les images à partir de la base de données de l'entreprise.

Kooaba [5] est une plateforme de reconnaissance d'images, qui permet aux utilisateurs de fournir leurs bases de données et de créer un système de reconnaissance d'images selon les bases de données fournies.

Ceci dit, on peut classer notre *Système avec paramètres adaptés pour chaque classe* parmi ces systèmes.

ANNEXE C

INTERFACE GRAPHIQUE DU SYSTÈME PROPOSÉ

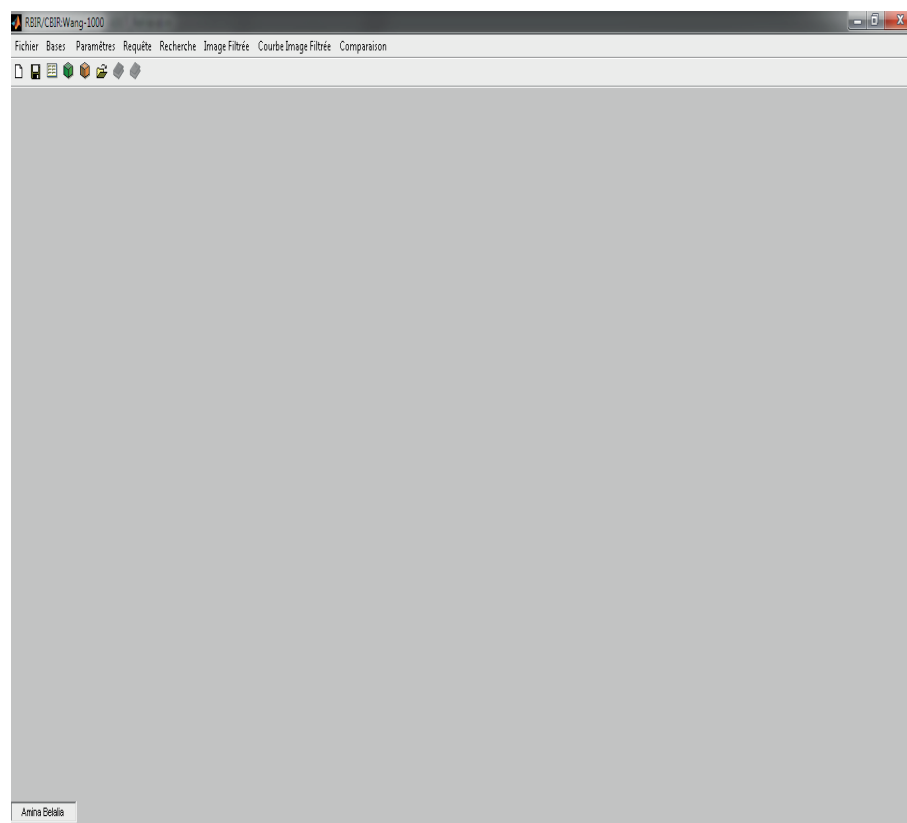


Figure 1 – Interface graphique de la méthode proposée.

EXEMPLES DE RECHERCHE PAR LE SYSTÈME CLASSIQUE (CBIR AVEC DCT SEULEMENT).

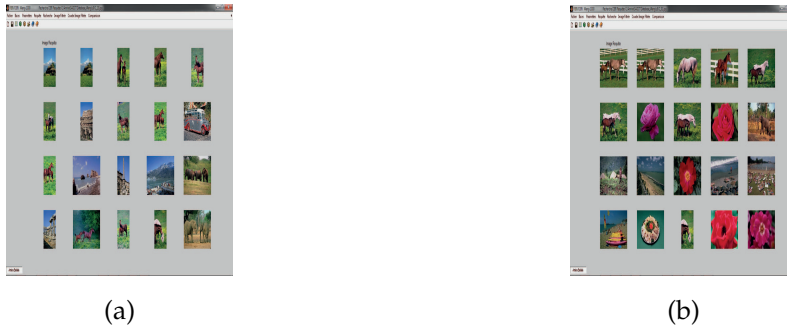


Figure 2 – Exemple de recherche par le système classique (CBIR avec DCT seulement) sur la base de données Corel-1000 : (a) Image montagne. (b) Image Cheval. La première image est l'image requête, les 19 autres images sont les images recherchées par le système classique (CBIR avec DCT seulement).

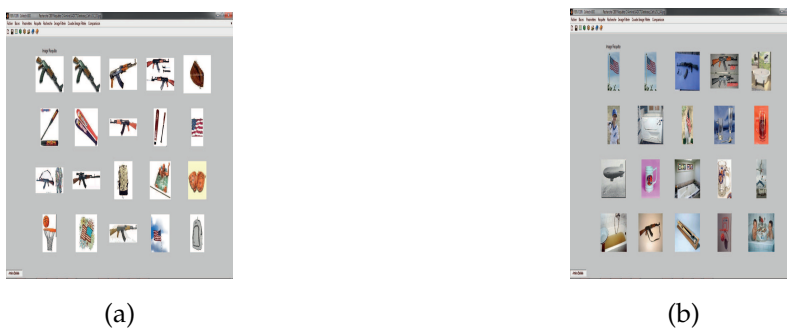


Figure 3 – Exemple de recherche par le système classique (CBIR avec DCT seulement) sur la base de données Caltech-256 : (a) Image AK-47. (b) Image Drapeau Américain. La première image est l'image requête, les 19 autres images sont les images recherchées par le système classique (CBIR avec DCT seulement) proposée.

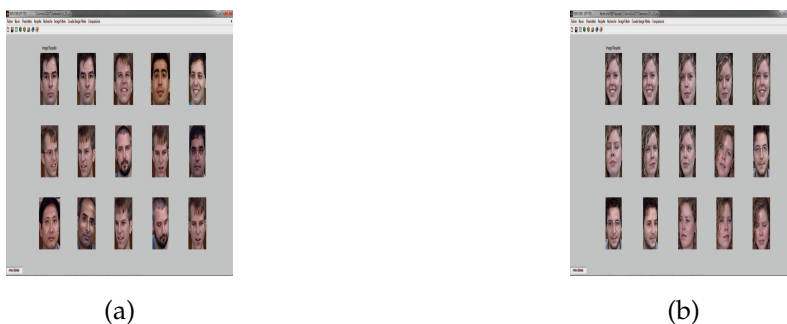


Figure 4 – Exemple de recherche par le système classique (CBIR avec DCT seulement) sur la base de données GTF : (a) Image Classe 1. (b) Image Classe 10. La première image est l'image requête, les 14 autres images sont les images recherchées par le système classique (CBIR avec DCT seulement) proposée.

EXEMPLES DE RECHERCHE PAR LE SYSTÈME AVEC PARAMÈTRES GLOBAUX (RBIR AVEC DCT ET SA-DCT).



Figure 5 – Exemple de recherche par le système avec paramètres globaux sur la base de données Corel-1000 : (a) Image montagne. (b) Image Cheval. La première image est l'image requête, les 19 autres images sont les images recherchées par le système avec paramètres globaux proposée.



Figure 6 – Exemple de recherche par le système avec paramètres globaux sur la base de données Caltech-256 : (a) Image AK-47. (b) Image Drapeau Américain. La première image est l'image requête, les 19 autres images sont les images recherchées par le système avec paramètres globaux proposée.

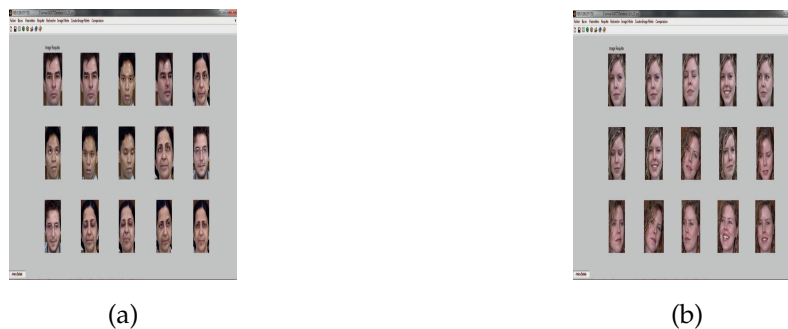


Figure 7 – Exemple de recherche par le système avec paramètres globaux sur la base de données GTF : (a) Image Classe 1. (b) Image Classe 10. La première image est l'image requête, les 14 autres images sont les images recherchées par le système avec paramètres globaux proposée.

EXEMPLES DE RECHERCHE PAR LE SYSTÈME AVEC PARAMÈTRES ADAPTÉS POUR CHAQUE CLASSE (RBIR AVEC DCT ET SA-DCT).

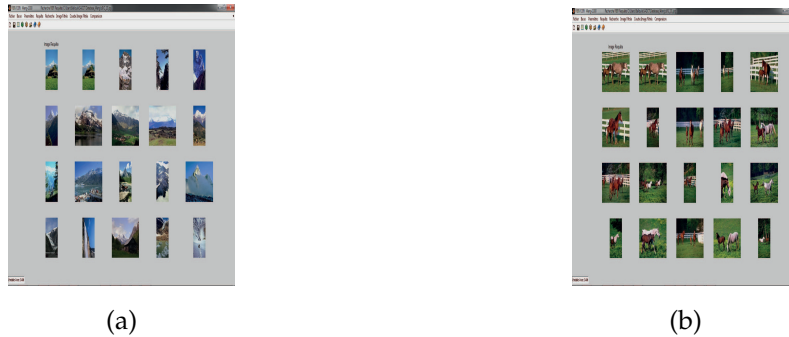


Figure 8 – Exemple de recherche par le système avec paramètres adaptés pour chaque classe sur la base de données Corel-1000 : (a) Image montagne. (b) Image Cheval. La première image est l'image requête, les 19 autres images sont les images recherchées par le système avec paramètres adaptés pour chaque classe proposée.

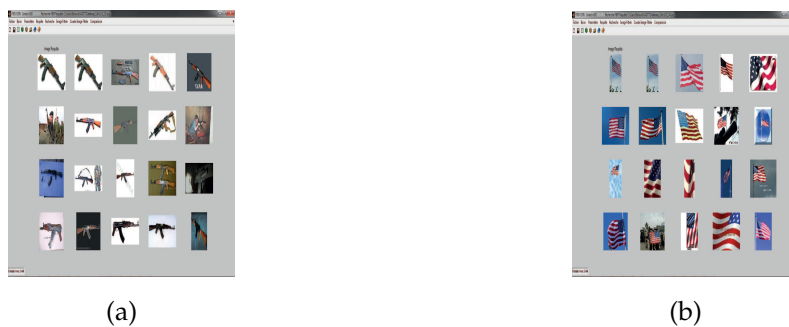


Figure 9 – Exemple de recherche par le système avec paramètres adaptés pour chaque classe sur la base de données Caltech-256 : (a) Image AK-47. (b) Image Drapeau Américain. La première image est l'image requête, les 19 autres images sont les images recherchées par le système avec paramètres adaptés pour chaque classe proposée.

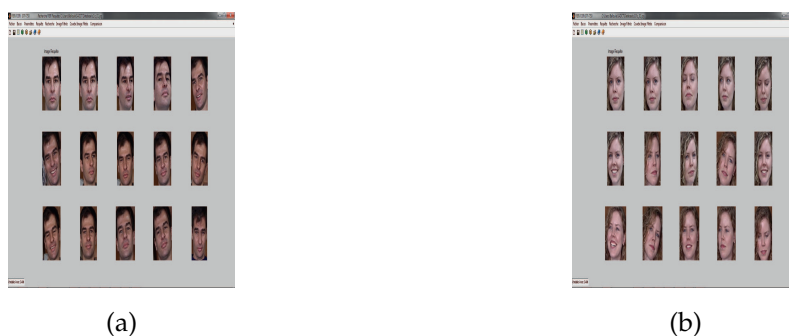


Figure 10 – Exemple de recherche par le système avec paramètres adaptés pour chaque classe sur la base de données GTF : (a) Image Classe 1. (b) Image Classe 10. La première image est l'image requête, les 14 autres images sont les images recherchées par le système avec paramètres adaptés pour chaque classe proposée.

LISTES DES PAPIERS

— Revues internationales avec comité de lecture

1. Kamel Belloulata, **Amina Belalia**, Shiping Zhu , "Object-Based Stereo Video Compression Using Fractals And Shape-Adaptive Dct", Int J Electron Commun(Aeü), July 2014.
2. **A. Belalia**, K Belloulata and K. Kpalma, " Region Based Image Retrieval using Shape-Adaptive DCT" Int J. Multimed. Info. Retr., Volume 4, pages 261-276, Dec 2015.
3. **A. Belalia**, K Belloulata and K. Kpalma, " Region Based Image Retrieval in compressed domain using Shape-Adaptive DCT" Multimedia Tools and Application, Accepted.

— Conférence internationales avec comité de lecture

1. K. Belloulata, L.Belhalouche, **A.Belalia**, K.Kpalma , "Region Based Image Retrieval Using Shape-Adaptive Dct", ChinaSIP IEEE, 2014, pp. 470-474.

— Conférence nationales avec comité de lecture

1. **Amina Belalia**, Kamel Belloulata, Lakhdar Belhalouche, "A New Region Based Image Retrieval In The Compressed Domain Using Shape-Adaptive DCT", Biomiec'14, Tlemcen, Oct 2014.
2. **Amina Belalia**, Kamel Belloulata, Kidiyo Kpalma, "Nouvelle Méthode d'Indexation d'Images par Région dans le Domaine Compressé Basée sur la SA-DCT", CITIM'15, Mascara, Algeria, April 2015.
3. **Amina Belalia**, Kamel Belloulata and Kidiyo Kpalma, "Nouvelle Méthode d'Indexation et de Recherche d'Image Basée-Région (RBIR) dans le Domaine Transformé Utilisant la SA-DCT.", SCCIBOV' 2015, SBA, Algeria.

BIBLIOGRAPHIE

- [1] "Compact Descriptors for Visual Search : Evaluation Framework", ISO/IEC JTC1 SC29 WG11 output document N12202, July 2011, .
- [2] "Test Model 2 : Compact Descriptors for Visual Search," ISO/IEC JTC1 SC29 WG11 output document W12734, April 2012, .
- [3] "White Paper on Compact Descriptors for Visual Search", ISO/IEC JTC1 SC29 WG11 output document N13951, April 2013, .
- [4] Like.com. 2011. "Wikimedia Foundation Inc." 29 sep 2012., <http://en.wikipedia.org/wiki/Like.com>.
- [5] Inbar, Shushu. "Kooba, Image Recognition." Kooaba.com. 2011, <http://www.kooba.com/>.
- [6] <http://www.vision.caltech.edu/image-databases/caltech256/>., Last accessed-March. 2014.
- [7] 3D. Special issue on 3-D video technology, IEEE Trans. Circuits Syst. Video Technol., vol. 10, no. 2-4, Mar-June 2000.
- [8] A.Natsev, R.Rastogi, et K.Shim. Walrus : A similarity retrieval algorithm for image database. Dans *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 395-406, 1999.
- [9] Bai, C., Kpalma, K., et Ronsin, J. Color textured image retrieval by combining texture and color features. Dans *Proceedings EUSIPCO'12 (European Signal Processing Conference)*, pages 170-174, 2012.
- [10] Bai, C., Kpalma, K., et Ronsin, J. A new descriptor based on 2d dct for image retrieval. Dans *Proceedings VISAPP'12 (International Conference on Computer Vision Theory and Applications)*, pages 714-717, Feb 2012.
- [11] Barnard, S.T. Stochastic stereo matching over scale. pages 769-778, Avril 1988.
- [12] BARNESLEY, M. F. *fractals every where*. Academic press, New York, 1988.
- [13] BARNESLEY , M. F., SLOAN A. D. A better way to compress images. *Byte*, pages 215-223, 1988.

- [14] Barthel, K., Schuttemeyer, J., Voye, T., et Noll, P. A new image coding technique unifying fractal and transform coding. Dans *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, pages 112–116, Oct 1994.
- [15] Belloulata, K. Fast fractal coding of subbands using a non-iterative block clustering. *Journal of Visual Communications and Image Representation*, 16(1), p. 55–67, 2005.
- [16] Belloulata, K., Baskurt, A., Benoit-Cattin, H., et Prost, R. Fractal coding of subbands with an oriented partition. *Signal Processing : Image Communication*, 12(3), p. 243–252, june 1998.
- [17] Belloulata, K. et Konrad, J. Fractal image compression with region-based functionality. *IEEE Transactions on Image Processing*, 11(4), p. 351–362, 2002.
- [18] Belloulata, K. et Rossignol, F. Compression fractales basée région des séquences vidéo segmentées. *colloque GRETSI'03 (19ème colloque international francophone de Traitement du Signal et de l'image)*, 12(3), p. 243–252, April 2003.
- [19] Belloulata, K., Stasiński, R., et Konrad, J. Region-based image compression using fractals and shape-adaptive DCT. Dans *Proceedings ICIP-99 (IEEE International Conference on Image Processing)*, volume II, pages 815–819, oct 1999.
- [20] Belloulata, K., Zhu, S., et Wang, Z. A fast fractal video coding using cross-hexagon search for block motion estimation. *International Scholarly Research Network*, 2011 (386128), p. 1–10, 2011.
- [21] B.Girod et al. Mobile visual search. Dans *IEEE Signal Processing Magazine*, volume 4, page 61–76, oct 2011.
- [22] Bolle, R.M., Pankanti, S., et Ratha, N.K. Evaluation techniques for biometrics-based authentication systems (frr). Dans *Proceedings ICPR'00 (International Conference on Pattern Recognition)*, volume II, pages 831–837, sep 2000.
- [23] Bracewell, R.N., Chang, K.Y., Wang, A.K., et Wang, Y.H. Affine theory for two-dimensional Fourier transform. *Electronics Letters*, 29(3), p. 304, 1993.
- [24] Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J., et Osher, S. Fast global minimization of the active contour/snake model. *J. Math. Imaging and Vision*, 28(2), p. 151–167, 2007.
- [25] Carson, C., Belongie, S., Greenspan, H., et Malik, J. Blobworld : Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8), p. 1026–1038, 2002.
- [26] Castagno, R., Ebrahimi, T., et Kunt, M. Video segmentation based on multiple features for interactive multimedia applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5), p. 562–571, 1998.

- [27] Chang, C., Chuang, J., et Hu, Y. Retrieving digital images from a jpeg compressed image database. *Image and Vision Computing*, 22(6), p. 471–484, 2004.
- [28] Chassaing, D., Choquet, B., et Pelé, D. A stereoscopic television system (3d-tv) and compatible transmission on a mac channel (3d-mac). 4, p. 33–43, 1991.
- [29] Chiang, C., Hung, Y., Yang, H., et Lee, G. Region-based image retrieval using color-size of watershed regions. *Journal of Visual Communications and Image Representation*, 20, p. 167–177, 2009.
- [30] database. <http://wang.ist.psu.edu/jwang/test1.tar> Last accessed-10 January 2013.
- [31] Datta, R., Joshi, D., Li, J., et Wang, J. Z. Image retrieval : Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2), p. 5 :1–60, 2008.
- [32] Domaszewicz, J. et Vaishampayan, V. A. Graph-theoretical analysis of the fractal transform. Dans *Proceedings ICASSP-95 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume IV, pages 2559–2562, apr 1995.
- [33] Douak, F., Benzid, R., et Benoudjit, N. Color image compression algorithm based on dct transform combined to adaptive block scanning. *International Journal of Electronics and Communications (AEÜ)*, 65, p. 16–26, 2011.
- [34] Edmundson, D., Schaefer, G., et Celebi, M. Robust texture retrieval of compressed images. Dans *Proceedings ICIP-12 (IEEE International Conference on Image Processing)*, volume IV, pages 2421–2424, oct 2012.
- [35] Feng, G. et Jiang, J. Jpeg compressed image retrieval via statistical features. *Pattern Recognition*, 36(4), p. 977–985, 2003.
- [36] Ferman, A., Tekalp, M., et Mehrotra, R. Robust color histogram descriptors for video segment retrieval and identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(5), p. 497–508, 2002.
- [37] Fisher, Y. *Fractal Image Compression with Quadrees : in Fractal Image Compression : Theory and Application*. Springer-Verlag, New York, 1995.
- [38] Fisher, Y., Shen, T. P., et Rogovin, D. Fractal (self-vq) encoding of video sequences. Dans *Proceedings VCIP-94 (Visual Communications and Image Processing)*, volume 2308, pages 1359–1370, 1994.
- [39] Franich, R., Lagendijk, R., et Biemond, J. Fractal coding in object-based system. Dans *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, pages 405–408, nov 1994.
- [40] Gall, D. Mpeg : A vidéo compression standard for multimédia applications. *Communications ACM*, 34(4), p. 47–58, Avril 1991.

- [41] Gilge, M., Engelhardt, T., et Mehlan, R. Coding of arbitrarily shaped image segments based on a generalized orthogonal transform. *Signal Processing : Image Communication*, 1(2), p. 153–180, Octobre 1989.
- [42] Griffin, G., Holub, A., et Perona, P. Caltech-256 object category dataset, 2007.
- [43] GTF Face database. <http://www.anefian.com/research/face-reco.htm>. Georgia Tech, GTF database. Last accessed-Mars 2010.
- [44] Jacquin, A. E. *A fractal theory of iterated Markov operators on spaces of measures with applications to digital image coding*. PhD thesis, Georgia Institute of technology, 1989.
- [45] Jacquin, A. E. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, 1(1), p. 18–30, 1992.
- [46] Jacquin, A. E. Fractal image coding : A review. *Proceedings of the IEEE*, 81(10), p. 1451–1465, 1993.
- [47] Jain, A.K. *Fundamentals of Digital Image Processing*. Information and System Sciences Series. Prentice Hall, New York, NY, 1989.
- [48] Jing, F., Li, M., Zhang, H., et Zhang, B. An efficient and effective region-based image retrieval framework. *IEEE Transactions on Image Processing*, 13(5), p. 699–709, 2004.
- [49] Kauff, P. et Schüür, K. An extension of shape-adaptive DCT (SA-DCT) towards DC separation and Δ DC correction. Dans *1997 Picture Coding Symposium*, pages 647–652, Septembre 1997.
- [50] Kherfi, M. L. *Rôle du contre-exemple dans le retour de pertinence en recherche d'images*. PhD thesis, University of Sherbrooke, Canada, 2002.
- [51] Kim, C. S., Kim, R. C., et Lee, S. U. Fractal coding of image sequence using extended circular prediction mapping. Dans *Proceedings EUSIPCO'98 (European Signal Processing Conference)*, pages 308–311, Sept 1998.
- [52] Kim, C. S., Kim, R. C., et Lee, S. U. Fractal coding of video sequence using circular prediction mapping and noncontractive interframe mapping. *IEEE Transactions on Image Processing*, 7(4), p. 601–605, 1998.
- [53] Labelle, L., Lauzon, D., Konrad, J., et Dubois, E. Arithmetic coding of a lossless contour-based representation of label images. Dans *Proceedings ICIP-98 (IEEE International Conference on Image Processing)*, volume I, pages 261–265, oct 1998.
- [54] Liu, Y., Zhang, D. S., et Lu, G. Region-based image retrieval with high-level semantics using decision tree learning. *Pattern Recognition*, 41(1), p. 2554–2570, 2008.

- [55] Liu, Y., Zhang, D. S., Lu, G., et Ma, W. Y. Study on texture feature extraction in region-based image retrieval system. Dans *Proceedings MMM'06 (International Multimedia Modeling Conference)*, pages 264–271, 2006.
- [56] Liu, Y., Zhang, D. S., Lu, G., et Ma, W. Y. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1), p. 262–282, 2007.
- [57] Liu, Y., Zhou, X., et Ma, W. Y. Extraction of texture features from arbitrary-shaped regions for image retrieval. Dans *Proceedings ICME'04 (International Conference on Multimedia and Expo)*, pages 1891–1894, june 2004.
- [58] Ma, W.Y. et Manjunath, B. S. Netra : A toolbox for navigating large image database. Dans *Proceedings ICIP-97 (IEEE International Conference on Image Processing)*, volume I, pages 568–571, Oct 1997.
- [59] Mandelbrot, B. Les objets fractales : formes, hasard et dimension. 1984.
- [60] Manipoonchelvi, P. et Muneeswaran, K. Significant region-based image retrieval. *Signal Image and Video Processing*, (6), p. –, 2014.
- [61] mpeg4. Special issue on representation and coding of images and video II, IEEE Trans. Circuit Syst. Video Technol., vol. 9, Dec. 1999, .
- [62] mpeg4. Special issue on representation and coding of images and video II, IEEE Trans. Circuit Syst. Video Technol., vol. 9, Dec. 1999, .
- [63] Murala, S., Maheshwari, R. P., et Balasubramanian, R. Directional local extrema patterns : a new descriptor for content based image retrieval. *International Journal of Multimedia and Information Retrieval*, 1(3), p. 191–203, 2012.
- [64] Naemura, T. et Harashima, H. Fractal coding of a multi-view 3-d image. Dans *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume III, pages 129–132, nov 1994.
- [65] Naemura, T. et Harashima, H. Self-similarity modeling for interpolation and extrapolation of multi-viewpoint image sets. Dans *Proceedings ICASSP-94 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume V, pages 369–372, apr 1994.
- [66] Oien, G. et Lepsoy, S. Fractal-based coding with fast decoder convergence. *Signal Processing*, 3, p. 105–117, 1994.
- [67] Salembier, P., Marqués, F., et al., M. Pardàs. Segmentation-based video coding system allowing the manipulation of objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1), p. 60–74, 1997.
- [68] Saupe, D. et Jacobs, S. Variance-based quadrees in fractal image compression. *Electronics Letters*, 31, p. 46–48, 1997.

- [69] Senapati, R. K., Pati, U. C., et Mahapatra, K. K. Listless block-tree set partitioning algorithm for very low bit rate embedded image compression. *International Journal of Electronics and Communications (AEÜ)*, 66, p. 985–995, 2012.
- [70] Sikora, T. Low complexity shape-adaptive DCT for coding of arbitrarily shaped image segments. *Signal Processing : Image Communication*, 7(4–6), p. 381–395, Novembre 1995.
- [71] Sikora, T. et Makai, B. Shape-adaptive DCT for generic coding of video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(1), p. 59–62, Février 1995.
- [72] Smeulders, A. W. M., Worring, M., et Santini, S. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), p. 1349–1380, 2000.
- [73] Stasinski, R. et Konrad, J. A new class of fast shape-adaptive orthogonal transforms and their application to region-based image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(1), p. 16–34, Feb. 1999.
- [74] Strintzis, M. G. et Malassiotis, S. Object-based coding of stereoscopic and 3d image sequences. *IEEE Transactions on Signal Processing*, 16(3), p. 14–28, 1999.
- [75] Vetro, A., Wiegand, T., et Sullivan, G. J. Overview of the stereo and multiview video coding : Extension of the h264/mpeg-4avc standard. *Proceedings of the IEEE*, 99(1), p. 626–642, 2011.
- [76] Wallace, K.K. The JPEG still picture compression standard. *Communications ACM*, 34(4), p. 30–44, Avril 1991.
- [77] Wang, J. Z., Li, J., et Wiederhold, G. Simplicity : Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9), p. 947–963, 2001.
- [78] Wang, R. S. et Wang, Y. Multiview video sequence analysis, compression, and virtual viewpoint synthesis. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(3), p. 397–410, 2000.
- [79] Wang, Y., Ostermann, J., et Zhang, Y. Q. *Video processing and communications*. Prentice-Hall, New York, 2002.
- [80] Wu, Y. T., Shih, F. Y., Shi, J., et Wu, Y. T. A top-down region dividing approach for image segmentation. *Pattern Recognition*, 41(1), p. 1948–1960, 2008.
- [81] Yang, W. et Ngi, N. K. Mpeg-4 based stereoscopic video sequences encoder. Dans *Proceedings ICASSP-04 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume V, pages 741–744, may 2004.

- [82] Yang, X. et Cai, L. Adaptive region matching for region-based image retrieval by constructing region importance index. *IET Computer Vision*, 8(2), p. 141–151, 2014.
- [83] Yanping, D. et Wang, J. Z. A scalable integrated region-based image retrieval system. volume I, pages 22–25, oct 2001.
- [84] Zhang, Y. et Wang, X. Fractal compression coding based on wavelet transform with diamond search. *Nonlinear Analysis : Real World Applications*, 13, p. 106–112, nov 2012.
- [85] Zhao, Y. et Yuan, B. A hybrid image compression scheme combining block-based fractal coding and dct. *Signal Processing : Image Communication*, 8, p. 73–78, 1996.
- [86] Zhong, D. et Defee, I. Dct histogram optimization for image database retrieval. *Pattern Recognition Letters*, 26(14), p. 2272–2281, 2005.
- [87] Zhong, D. et Defee, I. Performance of similarity measures based on histograms of local image feature vectors. *Pattern Recognition Letters*, 28(15), p. 2003–2010, 2007.
- [88] Zhong, D. et Defee, I. Face retrieval based on robust local features and statistical-structural learning approach. *EURASIP Journal on Advances in Signal Processing*, 2008 (ID 631297), p. 12 pages, 2008.
- [89] Zhu, S., Belloulata, K., et Belalia, A. Object-based fractal coding of video sequence. Dans *Proceedings of the International Workshop on Non Linear Signal and Image Processing*, may 2005.
- [90] Zhu, S., Hou, Y., Wang, Z., et Belloulata, K. Fractal video sequences coding with region-based functionality. *Applied Mathematical Modeling*, 36, p. 5633–5641, nov 2012.
- [91] Zhu, S., Tian, J., Shen, X., et Belloulata. A new cross-diamond search algorithm for fast block motion estimation. Dans *Proceedings ICIP-09 (IEEE International Conference on Image Processing)*, pages 1581–1584, nov 2009.
- [92] Zou, W., Kpalma, K., et Ronsin, J. Semantic image segmentation using region bank. Dans *Proceedings ICPR'12 (International Conference on Pattern Recognition)*, pages 922–925, Nov. 2012.
- [93] Zou, W., Kpalma, K., et Ronsin, J. Semantic segmentation via sparse coding over hierarchical regions. Dans *Proceedings ICIP-12 (IEEE International Conference on Image Processing)*, pages 2577–2580, Sept 2012.
- [94] Zou, W., Kpalma, K., et Ronsin, J. Automatic foreground extraction via joint crf and online learning. *Electronics Letters*, 49(18), p. 1140–1142, 2013.

Compression et Indexation Basée-Région d'Image dans le Domaine Transformé Utilisant la SA-DCT

Résumé :

Depuis de nombreuses années, l'image numérique est un outil favorisé dans des domaines très divers telles que la médecine, l'imagerie satellitaire et la télévision numérique. Dans ces domaines, lorsqu'on parle d'images et de séquences vidéo, ces sources d'informations brutes représentent une quantité de données très importantes et contiennent beaucoup de textures redondantes. D'où la nécessité de mise en place des méthodes d'encodage et de compression, avant la transmission ou le stockage, qui permettent de représenter la même information mais avec beaucoup moins de données. En plus, puis ce que les images prennent une part de plus en plus volumineuse, il apparaît important de concevoir des systèmes d'indexations, capables de mener à terme la recherche d'informations au sein de très grandes collections d'images numériques.

Dans cette thèse, deux nouveaux systèmes basés-régions et basés sur les transformations DCT et SA-DCT, qui sont largement utilisées dans le domaine de la compression, sont proposés. Le premier système est un système de codage de la vidéo qui permet de coder les objets dans les images de la séquence de façon indépendante et d'offrir plus de flexibilité sur le contrôle de la largeur de bande pour la transmission. Ce codec permet de mieux coder les frontières de différentes régions. Tandis que le deuxième système est un système d'indexation et de la recherche d'images basée région (RBIR) qui prend en considération une segmentation a priori de l'image en région, tel qu'il a été défini dans MPEG-4. Autrement dit, une image est représentée par des régions segmentées, et chaque région est associée à un vecteur de caractéristiques dérivé à partir de la DCT et la SA-DCT. Chaque région de l'image requête, peut être sélectionnée comme l'objet principal. Pour les images qui n'ont pas un objet principal, l'utilisateur peut sélectionner l'image entière comme requête.

L'objectif du travail proposé est la confirmation de l'efficacité des systèmes proposés par des résultats expérimentaux et des comparaisons par rapport aux systèmes déjà existants. Un système conjoint peut être réalisé à partir des deux systèmes proposés puisqu'il s'agit de deux systèmes qui ont le même objectif (basé-région) et qui travaillent avec les mêmes transformations DCT et SA-DCT.

Mots-clés : codage de la stéréo et de la vidéo, prédiction inter-vue, codage basé-région, le codage fractal, Quadtree, MPEG-4, Recherche d'images basé sur le contenu (CBIR), DCT, Segmentation, Recherche d'images basé-région (RBIR), SA-DCT.

Region-Based Image Compression and Retrieval in the Transformed Domain Using the SA-DCT

Abstract :

For many years, the digital image is a favored tool in very diverse fields such as medicine and satellite imaging or Digital TV. In these areas, when talking of images and video sequences, the raw information sources represent a very large amount of data and contain a lot of redundant textures. Hence necessitated the establishment of methods for encoding and compression prior to transmission or storage, which are used to represent the same information but with much less data. In addition, and that the images are playing an increasingly large, it seems important to design of indexing systems capable of completing the search for information in very large digital image collections.

In this thesis, two new systems based-region and based on DCT and SA-DCT transformation, which are widely used in the field of compression, are proposed. The first system is a video coding system for encoding objects in images of the sequence independently and offer more flexibility in the control of the bandwidth for transmission. This codec provides a better code on borders of different regions. Whereas the second system is an region based image retrieval (RBIR) which takes into account a prior segmentation of the image into regions, as it was defined in MPEG-4. In other words, an image is represented by segmented regions, and each region is associated with a feature vector derived from the SA-DCT and DCT. Each region of the query image, can be selected as the main object. For images that do not have a main object, the user can select the entire image as a query.

The objective of the proposed work is the confirmation of the effectiveness of systems from experimental results and comparisons to existing systems. A joint system can be made from the two systems and proposed that these two systems have the same goal (based-region) and work with the same transformations SA-DCT and DCT.

keywords : Stereo video coding, inter-view prediction, fractal coding, Region-based coding, Quadtree, MPEG-4, Content-Based Image Retrieval(CBIR), DCT, Segmentation, Region-Based Image Retrieval(RBIR), SA-DCT.

Cette thèse a été rédigée à l'aide du logiciel libre de mise en forme LATEX.