

# Représentation Compacte des Cartes de Segmentation dans les Séquences d'Images: étude comparative

par

**Jean-Bernard Chartier**

Ingénieur Télécom Paris

Institut National de la Recherche Scientifique  
16, Place du Commerce, Verdun, Québec, H3E 1H6

Mars 1995

Mémoire présenté comme exigence partielle de la  
maîtrise ès sciences en télécommunications

© Jean-Bernard Chartier, 1995



*A la mémoire de Coluche*



## Sommaire

L'objectif de cette recherche est de trouver, dans le contexte du codage vidéo à très bas débit, des représentations efficaces de cartes de segmentation pour les séquences d'images. Comme l'algorithme H.261 "basé-bloc" crée des artéfacts visuels majeurs dès que le débit est baissé, nous nous intéressons à d'autres techniques pour coder l'information vidéo à de très bas débits. Elles sous-entendent une analyse des séquences d'images de type "basé-région", axée sur le processus de segmentation de trames. Dans ce mémoire, une étude comparative de diverses techniques de compression de l'information de segmentation est proposée. Tout d'abord, les méthodes de codage sans perte: codage de la longueur des courses ("Run-Length Coding"), codage arithmétique binaire ("Binary Arithmetic Coding") et codage par chaînes ("Chain Coding") sont étudiées. Puis, une solution de codage avec perte, basée sur une approximation polygonale optimale des cartes de segmentation, est décrite. Les résultats obtenus mettent en valeur une amélioration du taux de compression; selon le modèle de codage sans perte utilisé, le nombre de bits par point de contour varie globalement de 4 à 1.3. Dans le cas d'un codage avec perte, ce taux approche la valeur 1, tout en conservant une qualité d'image subjectivement satisfaisante. De plus, il est nous possible de contrôler certains paramètres permettant d'établir un compromis entre la qualité visuelle et le taux de bits. Nous achevons l'étude en citant les orientations nouvelles à apporter au sujet pour en améliorer la rentabilité.

## Remerciements

Tout d'abord, je tiens à remercier mon professeur Janusz Konrad pour ses conseils, sa rigueur scientifique, son entière disponibilité et la confiance qu'il m'a témoignés tout au long de cette recherche. Je ne pourrais pas non plus oublier Albert Golembiowski pour ses conseils en matière de makefiles, de scripts GnuAwk et de programmation C++. Je souhaiterais aussi remercier Stéphane Coulombe pour les questions de bugs en C++, Christophe Stiller pour ses conseils pertinents en matière de codage à très bas débit, Abdol-Reza Mansouri pour sa contribution à la segmentation des images, Sylvain Fauvel pour les problèmes de système informatique, Viet-Nam Dang et toute l'équipe du Groupe Vidéo de l'INRS-Télécommunications.

Spéciale dédicace à tou(te)s mes ami(e)s punks de Montréal, San Francisco et Paris, à Véro et à mes parents.

# Table des matières

Sommaire	i
Remerciements	ii
Table des matières	iii
Liste des Figures	vi
Liste des Tableaux	xi
<b>1 Introduction</b>	<b>1</b>
<b>2 Survol de la littérature</b>	<b>5</b>
2.1 Débouchés du codage vidéo à très bas débit . . . . .	5
2.2 Envergure du standard MPEG-4 . . . . .	7
2.2.1 Recommandations définies pour MPEG-4 . . . . .	7
2.2.2 Avantages . . . . .	9
2.2.3 Standard courant H.261 pour le codage à bas débit . . . . .	9
2.2.4 Problèmes engendrés par un codage de type H.261 . . . . .	11
2.2.5 Suggestions et solutions de la littérature scientifique . . . . .	13
2.3 Intérêt de recherche à l'INRS sur le codage vidéo à très bas débit . .	17
2.3.1 Segmentation de l'image . . . . .	19

2.3.2	Représentation de la luminance: codage par transformée des textures de régions de forme arbitraire . . . . .	22
2.3.3	Codage des cartes de segmentation . . . . .	23
<b>3</b>	<b>Codage sans perte</b>	<b>25</b>
3.1	Description globale du codage des cartes de segmentation . . . . .	25
3.2	Codage des transitions horizontales . . . . .	27
3.2.1	Extraction des états de transition de cartes de segmentation . . . . .	29
3.2.2	Processus de reconstruction . . . . .	33
3.2.3	Codage effectif des états de transition . . . . .	36
3.2.4	Codage de la longueur des courses (RLC) . . . . .	37
3.2.4.1	Conclusion . . . . .	38
3.2.5	L'algorithme JBIG de codage arithmétique binaire . . . . .	39
3.2.5.1	La structure de JBIG . . . . .	40
3.2.5.2	Codage arithmétique binaire (BAC) . . . . .	42
3.2.5.3	Estimation de la probabilité . . . . .	47
3.2.5.4	Modèle de patron utilisé . . . . .	48
3.2.6	Conclusion . . . . .	51
3.3	Le codage par chaînes (CC) . . . . .	51
3.3.1	Description de l'algorithme CC . . . . .	52
3.3.2	Codage effectif des chaînes . . . . .	54
3.3.3	Codage du point de départ de chaque chaîne . . . . .	58
3.3.4	Reconstruction par remplissage intérieur . . . . .	59
3.3.5	Conclusions . . . . .	61
<b>4</b>	<b>Codage avec perte</b>	<b>65</b>
4.1	Enoncé de l'algorithme . . . . .	65



4.1.1	Formulation du problème et algorithme de programmation dynamique . . . . .	66
4.1.1.1	Formulation du problème . . . . .	66
4.1.1.2	Algorithme de Programmation Dynamique (PD) . . . . .	68
4.1.2	Extraction des sommets de la courbe polygonale . . . . .	68
4.1.3	Mesures de distorsion . . . . .	69
4.2	Codage de la chaîne approximée . . . . .	70
4.3	Reconstruction . . . . .	72
4.4	Conclusion . . . . .	72
<b>5</b>	<b>Résultats pour les séquences d'images</b>	<b>75</b>
5.1	Outils de simulation disponibles au sein du groupe vidéo de l'INRS . . . . .	75
5.2	Images de référence et types de segmentation . . . . .	75
5.3	Résultats pour la segmentation intra-trame . . . . .	76
5.3.1	Codage sans perte . . . . .	77
5.3.2	Codage avec perte . . . . .	80
5.4	Résultats pour la segmentation inter-trame . . . . .	84
5.4.1	Codage sans perte . . . . .	87
5.4.2	Codage avec perte . . . . .	89
5.5	Commentaires . . . . .	93
<b>6</b>	<b>Conclusion</b>	<b>97</b>
6.1	Travaux futurs . . . . .	99
	<b>Références</b>	<b>103</b>

# Table des figures

2.1	(a) Codage basé-bloc contre (b) codage basé-objet/région . . . . .	13
2.2	Exemple de diagramme d'un système de codage vidéo "basé-région" pour des applications à très bas débit . . . . .	19
2.3	Codage vidéo à très bas débit; rôle du codage des cartes de segmentation	24
3.1	Trame #0 des séquences (a) "Miss America", (b) "Carphone", (c) "Foreman" et un exemple de segmentations pour (d) "Miss America" (21 régions, PCM: 127 kbits), (e) "Carphone" (74 régions, PCM: 177 kbits), (f) "Foreman" (91 régions, PCM: 177 kbits) obtenues par la méthode basée sur MDL . . . . .	26
3.2	Codage/décodage des cartes de segmentation . . . . .	28
3.3	Principe de codage des cartes de segmentation, en utilisant les états de transition horizontale . . . . .	29
3.4	Exemples d'attribution des mots-codes 2 et 3 . . . . .	31
3.5	Exemple d'une (a) segmentation avec (b) différents états de transition	32
3.6	Processus de reconstruction basé sur la cohérence d'étiquetage . . . . .	35
3.7	(a) Un codeur JBIG peut être décomposé en une chaîne de (b) $D$ codeurs différentiels et d'un (c) codeur en queue . . . . .	41
3.8	Fission de l'intervalle pour $N_{sk} = 4$ , (a) avec (T) et $sk = 3$ , (b) avec (T) et $sk = 1$ . . . . .	44
3.9	Organigramme du codeur arithmétique binaire . . . . .	45

3.10	Organigramme du décodeur arithmétique binaire . . . . .	46
3.11	Diagramme d'état de l'estimateur de probabilité JBIG . . . . .	47
3.12	Diagramme d'état de notre unité statistique . . . . .	48
3.13	Modèle de patron pour la décision des $\mathcal{S}$ avec le balayage proposé. En gris est représenté le pixel pour qui le $\mathcal{S}$ est à déterminer. . . . .	49
3.14	Exemple d'une (a) segmentation avec (b) les différents états de transition associés au codage BAC . . . . .	50
3.15	Exemple de BAC: (a) une trame segmentée, (b) extraction des états de transition et (c) représentation binaire des $\mathcal{S}$ . . . . .	50
3.16	Modèles de voisinage (a) par quatre pour l'extraction des contours, (b) par huit pour le suivi des contours extraits . . . . .	52
3.17	Exemple d'attribution des directions dans le "chain coding" . . . . .	54
3.18	Exemple de codage de contour par la méthode CC: (a) carte de segmentation, (b) extraction des contours et (c) codage des bouts de chaînes avec l'ordre A,C,B . . . . .	55
3.19	Attribution des codes S, E et I dans le processus de reconstruction . . .	60
3.20	Exemple de reconstruction: (a) décodage des directions, (b) attribution des codes S, E et I et (c) processus de reconstruction complété . . . .	62
4.1	Mesures d'erreur: (a) $\Delta(i, j) = D$ (ME) et (b) $\Delta(i, j) = D1^2 + D2^2 +$ $D3^2$ (MSE) entre deux sommets $M_i$ et $M_j$ . . . . .	70
4.2	"Miss America" #90: (a) trame originale, (b) segmentation intra, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs d'approximation (d) $\gamma = 7$ , (e) $\gamma = 10$ et (f) $\gamma = 13$ . . . . .	73
5.1	Séquences de référence (a) "Miss America", (b) "Carphone" et (c) "Foreman" . . . . .	76

5.2	Exemple de segmentations intra-trame de (a) “Miss America” #105, (b) “Carphone” #123, et (c) “Foreman” #180 obtenues par la méthode basée sur MDL . . . . .	77
5.3	Nombre de bits par intra-trame pour la séquence “Miss America” . . .	78
5.4	Nombre de bits par intra-trame pour la séquence “Carphone” . . . . .	79
5.5	Nombre de bits par intra-trame pour la séquence “Foreman” . . . . .	79
5.6	Nombre de bits par intra-trame pour la séquence “Miss America”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	81
5.7	Mesure de distorsion intra-trame pour la séquence “Miss America”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	81
5.8	Nombre de bits par intra-trame pour la séquence “Carphone”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	82
5.9	Mesure de distorsion par intra-trame pour la séquence “Carphone”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	82
5.10	Nombre de bits par intra-trame pour la séquence “Foreman”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	83
5.11	Mesure de distorsion par intra-trame pour la séquence “Foreman”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	83
5.12	Evolution des bits avec la mesure de distorsion pour les trois séquences	84
5.13	“Miss America” #135: (a) trame originale, (b) segmentation intra, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d) $\gamma = 7$ , (e) $\gamma = 10$ et (f) $\gamma = 13$ . . . . .	85
5.14	“Carphone” #159: (a) trame originale, (b) segmentation intra, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d) $\gamma = 7$ , (e) $\gamma = 10$ et (f) $\gamma = 13$ . . . . .	85

5.15	“Foreman” #189: (a) trame originale, (b) segmentation intra, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d) $\gamma = 7$ , (e) $\gamma = 10$ et (f) $\gamma = 13$ . . . . .	86
5.16	Exemple de segmentations inter-trame de (a) “Miss America” #6, (b) “Carphone” #3, et (c) “Foreman” #150 obtenues par la procédure “basé-mouvement” . . . . .	86
5.17	Nombre de bits par inter-trame pour la séquence “Miss America” . . .	87
5.18	Nombre de bits par inter-trame pour la séquence “Carphone” . . . . .	88
5.19	Nombre de bits par inter-trame pour la séquence “Foreman” . . . . .	88
5.20	Nombre de bits par inter-trame pour la séquence “Miss America”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	89
5.21	Taux de distorsion par inter-trame pour la séquence “Miss America”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	90
5.22	Mesure de bits par inter-trame pour la séquence “Carphone”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	90
5.23	Mesure de distorsion par inter-trame pour la séquence “Carphone”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	91
5.24	Nombre de bits par inter-trame pour la séquence “Foreman”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	91
5.25	Mesure de distorsion par inter-trame pour la séquence “Foreman”, avec les facteurs $\gamma = 7, 10, 13$ . . . . .	92
5.26	Evolution des bits avec le taux de distorsion pour les trois séquences	93
5.27	“Miss America” #6: (a) trame originale, (b) segmentation inter, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d) 10, (e) 13 et (f) 16 . . . . .	94

5.28	“Carphone” #120: (a) trame originale, (b) segmentation inter, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d) 10, (e) 13 et (f) 16 . . . . .	94
5.29	“Foreman” #150: (a) trame originale, (b) segmentation inter, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d) 10, (e) 13 et (f) 16 . . . . .	95

# Liste des tableaux

3.1	Table de Huffman associée à la Fig. 3.3 ( $\hat{H} = 0.65, \hat{L} = 1.15$ ) . . . . .	36
3.2	Tables de Huffman utilisées pour “Miss America” ( $\hat{H} = 0.15, \hat{L} = 1.02$ ), “Carphone” ( $\hat{H} = 0.28, \hat{L} = 1.05$ ) et “Foreman” ( $\hat{H} = 0.36, \hat{L} = 1.07$ ). Distributions évaluées sur les séquences segmentées . . . . .	37
3.3	Tables de Huffman en codant conjointement deux symboles, pour “Miss America” ( $\hat{H} = 0.3, \hat{L} = 1.07$ ), “Carphone” ( $\hat{H} = 0.54, \hat{L} = 1.14$ ) et “Foreman” ( $\hat{H} = 0.72, \hat{L} = 1.20$ ). $\hat{H}$ et $\hat{L}$ sont ici exprimés en bits par deux symboles joints. Les distributions ont aussi été évaluées sur les séquences segmentées . . . . .	37
3.4	Partie de dictionnaire du RLC pour $m = 16$ . . . . .	39
3.5	Nombre de bits, pour la trame #0 segmentée de chaque séquence, par codage RLC . . . . .	39
3.6	Nombre de bits, pour la trame #0 segmentée de chaque séquence, par codage arithmétique binaire simplifié (BAC) . . . . .	51
3.7	Histogramme (%) des états conditionnels $X_n X_{n-1}$ pour “Miss America” ( $\hat{H}_{co} = 1.07$ ) . . . . .	58
3.8	Histogramme (%) des états conditionnels $X_n X_{n-1}$ pour “Carphone” ( $\hat{H}_{co} = 1.16$ ) . . . . .	58
3.9	Histogramme (%) des états conditionnels $X_n X_{n-1}$ pour “Foreman” ( $\hat{H}_{co} = 1.17$ ) . . . . .	59

3.10	Nombre de bits, pour la trame #0 segmentée de chaque séquence, par codage CC . . . . .	61
4.1	Création de quatre classes de chaînes à coder . . . . .	71
4.2	Codage avec perte de “Miss America” #0, “Carphone” #0 et “Foreman” #0 pour la segmentation intra-trame . . . . .	73
5.1	Nombre moyen de régions et nombre moyen des points de contour (cp) pour les trois séquences traitées en mode intra . . . . .	77
5.2	Codage sans perte des segmentations intra-trame. (*) = sans compter le codage des points de départ des chaînes . . . . .	78
5.3	Codage avec perte des segmentations intra-trame . . . . .	80
5.4	Nombre moyen de régions et nombre moyen des points de contour (cp) pour les trois séquences traitées en mode inter . . . . .	86
5.5	Codage sans perte des segmentations inter-trame. (*) = sans compter le codage des points de départ des chaînes . . . . .	87
5.6	Codage avec perte des segmentations inter-trame . . . . .	89



# Chapitre 1

## Introduction

La compression vidéo reste depuis des années l'un des grands centres d'intérêt de la recherche vidéo. Sa standardisation est devenue nécessaire, car seul un standard réduit le coût élevé des codecs (codeur-décodeur) de compression vidéo et résout le problème contraignant d'inter-compatibilité des équipements réalisés par différentes entreprises. Ce développement a réellement débuté avec la mise en place des standards H.261, MPEG-1 et MPEG-2 proposant des compressions de l'information vidéo à taux fixes et qualités variables. Chaque standard couvre une gamme spécifique de débits de transmission. En fait, la conception du standard MPEG-2 est en phase finale tandis que celle des deux premiers cités est achevée depuis quelques années; des circuits intégrés, pour le codage et décodage de MPEG-1 à 1,5 Mbits/s, de MPEG-2 à des débits inférieurs à 20 Mbits/s [1] et de H.261 à  $p \times 64$  kbits/s [2] commencent à apparaître sur le marché.

A l'heure actuelle, un nouveau challenge stimule la communauté scientifique de chercheurs en compression vidéo. Il est le fruit de l'émergence d'applications vidéo telles que la vidéoconférence ou le visiophone, ou de celles liées aux communications sans fil ("wireless"), aux réseaux ATM qui permettraient de faire passer un grand

nombre de services sur un même canal. Désormais, il est question de développer une technologie pour la transmission des séquences vidéo à très bas débit, c'est à dire, pour des taux inférieurs à 64 kbits/s. Ce projet est défini dans le cadre des activités MPEG (Motion Picture Expert Group) [3]. Plus précisément, le développement d'un standard international pour le codage générique audio-vidéo à très bas débit constitue le nouveau centre d'intérêt de la recherche et devrait aboutir à ce que l'on désigne déjà par le standard MPEG-4. La nature générique du travail s'inscrit dans la volonté d'utiliser ce standard pour un grand nombre d'applications aussi diversifiées que possible. Comme les anciennes techniques, type codage hybride DCT+DPCM avec compensation de mouvement, appliquées à de bas débits, présentent de très faibles performances et démontrent clairement leurs limites, de toutes nouvelles solutions sont nécessaires pour l'établissement du standard MPEG-4. Un important effort international doit être accompli pour arriver au terme de cette recherche, justifiant le fait que plusieurs sociétés et instituts de recherches se soient impliqués dans le projet. De plus, un grand nombre de marchés potentiels et d'importants profits sont à prévoir, cette recherche constitue a fortiori un pôle d'attraction prioritaire.

Récemment, le groupe des Communications Visuelles de l'INRS-Télécommunications s'est engagé dans une activité de recherche portant sur le codage des séquences vidéo à très bas débit. La recherche, actuellement dans ses prémices, introduit la conception du codage "basé-région" ou "basé-objet", contrairement aux techniques traditionnelles de codage vidéo de type "basé-bloc". Cette notion provient de l'analyse de l'image faite au niveau du transmetteur, qui identifie les régions d'intensité uniforme ou à mouvement homogène. Après extraction, de telles régions sont typiquement traitées par une prédiction compensée par le mouvement, puis par des transformations orthonormales appliquées à des formes arbitraires. Ensuite, les cartes de

segmentation, représentant une partition de l'image en régions cohérentes, ainsi que les paramètres du mouvement de chaque région, doivent être transmis. Tous ces éléments seront discutés au Chapitre 2.

Ce mémoire traite du problème de la compression des cartes de segmentation; c'est à dire le codage à de très bas débit de séquences d'images segmentées selon un procédé "basé-région", où chaque étiquette de l'image coïncide avec une région. Pour ce faire, nous allons exposer successivement différentes techniques, soit entièrement inspirées de la littérature scientifique, soit modifiées par une contribution personnelle. Les diverses méthodes présentées assurent une bonne compression de l'information de segmentation. Partant d'une description de codages sans perte à un algorithme de codage avec perte, l'étude démontre une évolution positive dans la qualité des résultats obtenus. De plus, les avantages, inconvénients et améliorations possibles de chaque famille de techniques seront analysés.

Nous proposons le plan suivant: dans une première étape, un bilan de la littérature associée au sujet est effectué. Il inclut les principales applications désirées de cette recherche, les spécifications et recommandations sur le standard MPEG-4, ainsi que l'état actuel des investigations entreprises. Ensuite, nous décrirons trois méthodes de compression sans perte: codage de la longueur des courses (RLC), codage arithmétique binaire (BAC) et codage par chaînes (CC). Un codage avec perte des cartes de segmentation est ensuite proposé, combinant la technique sans perte de codage CC avec un algorithme d'approximation optimale polygonale des contours. Dans le chapitre suivant, les résultats obtenus après traitement de trois séquences de référence sont présentés et interprétés. Il nous semble important de préciser, qu'à titre de comparaison, le taux défini par le nombre bits par point de contour atteint 1.3 pour le meilleur codage sans perte. De plus, dans une étape ultérieure, grâce à

l'implantation d'un algorithme de programmation dynamique pour le codage avec perte, ce taux est réduit sous 1. Nous concluons en citant les nouvelles perspectives et orientations envisagées pour cette recherche, en vue d'améliorer l'efficacité de la compression des cartes de segmentation.

# Chapitre 2

## Survol de la littérature

Dans l'introduction, nous avons insisté sur l'importance actuelle du codage vidéo à très bas débit qui résulte directement d'une demande récente et croissante d'applications liées aux communications sans fil et aux réseaux ATM. Sachant que les anciens algorithmes ne satisfont pas les exigences requises pour de telles applications, un nouveau standard est attendu par les fabricants pour permettre la réalisation des produits et leur diffusion sur le marché.

### 2.1. Débouchés du codage vidéo à très bas débit

Les techniques de codage audio-vidéo à très bas débit vont plus particulièrement être développées pour les applications à capacité restreinte concernant la transmission et le stockage. Cette notion de limitation de capacité, ou capacité restreinte, est une caractéristique principale des médias interactifs, tels que les cartes-mémoires ou "smart cards", et des systèmes à très large distribution audio-vidéo, tels que le mail électronique ou les bases de données audio-vidéo. Les applications potentielles qui

requièrent de tels réseaux ou fonctions incluent:

- *le vidéophone*: cette application devrait fournir un service de vidéophonie capable de satisfaire les attentes des usagers auxquelles la technologie actuelle est loin de répondre [4],
- *le mail électronique multimédia* devrait fournir un système de messagerie maniable, ne dépendant pas du média ou du type de réseau. Les supports traditionnels, ainsi que les nouveaux, profiteront de ce standard pour le stockage et la transmission d'importantes quantités d'information,
- *la télésurveillance* inclut le transport unilatéral de l'information audio-vidéo entre emplacements éloignés, technique développée par exemple dans tous les systèmes de sécurité,
- *les bases de données multimédia interactives*: cette application requiert un accès interactif aux bases de données multimédia, sur une large variété de réseaux à bande étroite et de médias à capacité de stockage restreinte (images, séquences d'images en mouvement, imagerie par ordinateur, parole, ...),
- *imagerie interactive par ordinateur*: les systèmes de réalité virtuelle proposent un voyage interactif dans un environnement audio-vidéo généré par des procédés conventionnels de programmation. Plusieurs applications peuvent ainsi être réalisées à l'aide d'interfaces utilisatrices et de moniteurs faible résolution.
- *le "captioning"* consiste en la réalisation de programmes éducatifs pour sourds, mal-entendants ou personnes handicapées, et la conception d'utilitaires pour l'insertion de légendes dans les programmes multimédia conventionnels de toutes sortes,
- *les jeux*: amélioration des jeux portables actuellement sur le marché, ainsi que des jeux TV interactifs.

---

## 2.2. Envergure du standard MPEG-4

MPEG a produit deux importants standards (MPEG-1 et MPEG-2) et entamé son action dans le développement du standard MPEG-4, dont l'aboutissement est prévu en 1998 [5]. L'équipe MPEG prévoit un certain nombre d'étapes, l'une d'entre elles étant le regroupement nécessaire des propositions ou des solutions soumises par les parties intéressées en présence, sur la base de spécifications et de recommandations récemment établies. Pour résumer, MPEG-4 est un standard de codage audio-vidéo qui supporte de nouveaux moyens de communication, d'accès et de manipulation des données numériques. Grâce aux opportunités offertes par les faibles coûts de production et d'exploitation, les performances high-tech, et le défi croissant des bases de données multimédia, MPEG-4 devrait offrir une structure souple et un ensemble d'outils couvrant à la fois les anciennes et nouvelles fonctionnalités. En d'autres termes, MPEG-4 est sensé construire un système générique de codage audio-vidéo pour la transmission et le stockage à capacités restreintes. Il serait utilisé pour fournir, aussi bien aux industries qu'au grand public, les applications précédemment citées.

### 2.2.1. Recommandations définies pour MPEG-4

Comme pour tout standard, l'établissement de MPEG-4 est basé sur un ensemble de spécifications et de recommandations que nous allons rappeler dans la suite du texte; il est certain que quelques-unes d'entre elles sont totalement utopiques, si l'on tient compte du temps alloué pour la réalisation de certaines fonctionnalités. En conséquence, dans le cas du codage audio-vidéo à très bas débit, MPEG a identifié le besoin d'une première solution à court terme et d'une seconde plus élaborée à

long terme.

- Tout d’abord, les réseaux et médias de stockage envisagés pour les applications sollicitent des débits allant de 4.8 à 64 kbits/s.
- On prévoit d’utiliser un algorithme de codage vidéo de type H.261 (hybride DCT+DPCM suivi d’une compensation par le mouvement), mais entièrement optimisé pour le très bas débit.
- De plus, on souhaite que les systèmes de codage vidéo assurent une qualité visuelle acceptable. Ce standard doit pouvoir fournir une bonne représentation du mouvement, et améliorer la résolution spatiale au prix d’une perte de rendement du mouvement. Il s’avère que des efforts particuliers doivent se concentrer sur ce point, car aucun système ne produit encore une telle qualité subjective exigée par le très bas débit.
- Ceci présuppose une implantation aisée en temps réel des codeur et décodeur désirés pour la gamme d’applications. Comme la contrainte “en temps-réel” n’est pas forcément réalisable, un délai, n’excédant pas 150 *ms*, serait toléré.
- Le standard doit établir une syntaxe et un plan de codage pour supporter la manipulation facile et l’édition du flux de bits, sans faire appel à un transcodage.
- L’efficacité du standard doit être optimale pour les formats d’image QCIF et le traitement des images fixes haute résolution doit être effectué par le standard JPEG.
- MPEG-4 doit fournir des moyens efficaces pour l’accès aléatoire à très bas débit.
- De puissantes techniques génériques de corrections d’erreur sont nécessaires pour la plupart des applications (réseaux sans fil et stockage des données).



- 
- Comme cela a déjà été signalé, le standard doit être générique, indépendant du canal de transmission, du média utilisé et de l'application considérée, mais plutôt axé sur la forme du flux de bits. En quelque sorte, il doit posséder les caractéristiques d'un standard universel.
  - Pour finir, mentionnons les fonctionnalités impliquant le traitement de multiples flux de données, le développement d'outils multimédia pour l'accès aux données, le problème de la synchronisation, la gestion de données auxiliaires, la souplesse pour l'allocation d'un canal virtuel, des moyens de contrôle pour l'utilisateur, le traitement simultané de médias, des clés de sécurité...

### 2.2.2. Avantages

Les avantages liés à la réalisation de ce projet sont les suivants:

- création de nouveaux services high-tech audio-vidéo,
- une meilleure gestion de la largeur de bande des canaux de transmission à bande étroite,
- une meilleure gestion du stockage des divers médias, spécialement pour les systèmes à mémoire et capacité restreintes,
- ce codage générique doit fournir une certaine universalité des applications électroniques. Axé sur de faibles coûts, il devrait faciliter l'échange d'informations audio-vidéo entre différents domaines d'applications,
- une robustesse algorithmique.

### 2.2.3. Standard courant H.261 pour le codage à bas débit

Si l'on se réfère aux recommandations faites par MPEG, le standard MPEG-4 doit s'inspirer de la norme H.261.

La recommandation internationale H.261 “Video Codec for Audiovisual Services at  $p \times 64$  kbits/s” approuvée en 1990 s’adresse à la vidéophonie et la vidéoconférence. En conséquence, cet algorithme de codage vidéo doit fonctionner en temps réel, et ne peut donc tolérer qu’un délai minimal. Les formats d’image adoptés par CCITT sont les “Common Intermediate Format” (CIF:  $352 \times 288$ ) et “Quarter-CIF” (QCIF:  $176 \times 144$ ). Ces deux formats sont divisés en une structure hiérarchique de blocs, nécessaire pour un codage vidéo à forte compression: l’image, le groupe de blocs (GOB), les macro-blocs (MB) et les blocs. Chaque MB est constitué de quatre blocs  $8 \times 8$  de luminance (Y), et de deux blocs  $8 \times 8$  de chrominance ( $C_B$  et  $C_R$ ). Une image QCIF possède  $9 \times 11$  MB.

L’objectif principal du codage vidéo est de réduire au maximum le taux de transmission, en éliminant l’information redondante dans les séquences d’image. L’algorithme H.261 utilise le codage intra-trame et inter-trame; plus précisément, il s’agit d’un codage hybride d’une DCT (Discrete Cosine Transform) et d’une DPCM (Differential Pulse Code Modulation) avec compensation de mouvement. En effet, la prédiction s’appuie sur l’estimation de mouvement, d’une précision d’un pixel, en comparant chaque MB (généralement seulement la luminance) de la trame courante à sa version décalée dans la trame précédente. Il s’agit du principe d’appariement de blocs. La différence est transformée par DCT, quantifiée (en utilisant RLC sur un balayage en zig-zag) et transmise, avec l’information du mouvement, au multiplexeur vidéo. Pour augmenter l’efficacité de ce codage, un codage d’entropie, à longueur variable de mots-codes (VLC), est appliqué dans le multiplexeur vidéo qui suit immédiatement le codeur de source. Une table de type Huffman est utilisée [2].

Ce type de codage, légèrement remanié, pourrait s’avérer intéressant pour le très

bas débit. Bien que les raisons d'étendre celui-ci soient nombreuses, de multiples inconvénients limitent cependant son application au codage vidéo à très bas débit. Les problèmes majeurs rencontrés font l'objet de la section suivante.

#### **2.2.4. Problèmes engendrés par un codage de type H.261**

Il a été mentionné dans la section 2.2.1 que l'un des critères fondamentaux de MPEG-4 est son aptitude à assurer une qualité visuelle acceptable. Seulement, si l'on fait fonctionner les standards existants à de très bas débits, le compromis indispensable entre les résolutions spatiale et temporelle cause de nombreux artéfacts spatiaux, et une nette dégradation visuelle du mouvement apparaît. Ceci paraît normal puisque H.261 n'avait pas été développé pour de telles contraintes. Il est évident que la qualité d'image se détériore avec une diminution de débit. De plus, dès que les hypothèses principales faites sur les propriétés du mouvement dans les séquences d'image ne sont plus satisfaites (intensité constante le long des trajectoires du mouvement, mouvement linéaire, effets d'occlusion non pris en compte), les problèmes sont amplifiés.

Précisons les limitations de l'approche H.261; puisque cette norme s'appuie sur l'utilisation de blocs pour la compression de l'information vidéo, ce codage 2-D orienté-bloc souffre d'artéfacts visuellement contraignants (effet de bloc, "mosquito noise"). Le nombre fixe de paramètres de mouvement par bloc limite le taux de compression. De plus, les problèmes d'occlusion sont loin d'être résolus: en effet, les défauts spatiaux près de contours et les aires d'occlusion sont visibles après décodage. Finalement, ce type de codage n'est pas du tout adapté à la description et à la représentation de la réalité; il n'en propose qu'une grossière approximation. L'explication suivante peut être avancée: dans une séquence, les différences entre trames sont principalement causées par le déplacement des objets, mais aussi par les

zooms de la caméra, le bruit intrinsèque du matériel vidéo, des effets d'éblouissement, des changements de forme et de luminance des objets, des occlusions d'objets, etc., ces changements ne s'effectuant pas obligatoirement de bloc en bloc. En outre, l'étape de prédiction peut être sous-optimale, puisque la taille du bloc de prédiction ( $16 \times 16$ ) peut s'avérer être trop grande et/ou la fenêtre de recherche trop petite.

Ces différentes raisons ont conduit à développer une autre méthode s'appuyant sur un type de codage vidéo bidimensionnel basé-objet (Fig. 2.1); ce codage requiert une transmission d'information des contours, du mouvement, de la texture de chaque objet. Les recherches sont menées depuis peu d'années et quelques articles publiés confirment l'intérêt à consacrer à cette nouvelle technique de codage. Nous allons présenter les résultats les plus significatifs dans la prochaine section. Néanmoins, dans [6], il est fait état de plusieurs remarques négatives portant sur le codage 2-D basé-objet. Il apparaît que l'ancienne méthode d'extraction des objets (un seuillage grossier de la différence entre deux trames consécutives) fait défaut; la traînée, région bruitée près des contours d'objets en mouvement, est introduite pendant la reconstruction de l'image, quand de faux contours sont créés pendant l'extraction. De plus, cette méthode ne peut fonctionner sur des scènes plus compliquées que la classique image "head-and-shoulders", dans lesquelles le fond et l'avant-plan ne sont plus séparés. Cette méthode n'autorise la présence que d'un seul objet. Des problèmes similaires sont rencontrés quand l'image est fortement bruitée.

Dans la suite, nous allons énumérer quelques suggestions justifiant l'intérêt du codage "basé-objet" pour le très bas débit.

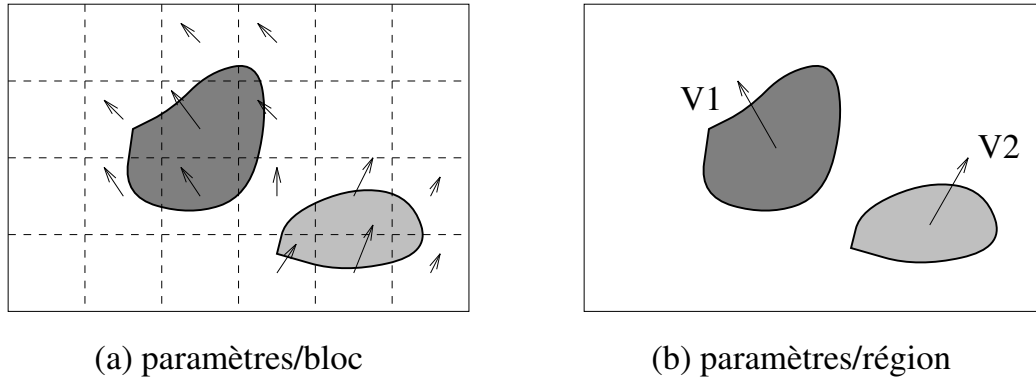


FIG. 2.1 - (a) Codage basé-bloc contre (b) codage basé-objet/région

### 2.2.5. Suggestions et solutions de la littérature scientifique

Pour palier au manque de techniques pour le codage vidéo à très bas débit, deux approches peuvent être distinguées. La première est proposée par des entreprises qui, dans le cadre de la norme H.26p, tentent d’apporter des améliorations significatives au standard H.261. Comme ces moyens classiques de codage n’ont pas encore convaincu de leur grande efficacité au très bas débit, une seconde approche, impliquant certains centres de recherche vidéo, est mise en œuvre: une solution nouvelle doit être proposée pour fournir une qualité audio-vidéo acceptable pour une gamme étendue d’applications à très bas débit. Ceci devrait démontrer des progrès importants fournis par un codage “basé-région” par rapport aux performances des standards actuels de type “basé-bloc”. Les nouveaux concepts de codage précédemment cités seront pris en compte pour obtenir toutes les améliorations envisagées. Malheureusement, les nouvelles techniques à mettre en œuvre s’avèrent très complexes et sont tributaires d’un délai de codage substantiel. Certes l’information du mouvement se traduit par un gain du codage, mais en contrepartie l’augmentation du coût est notable.

1. *Quelques suggestions liées à l’extraction d’objet.*

Les auteurs de [6] affirment que l’utilisation de l’information du mouvement

devrait résoudre un grand nombre de difficultés. Tout d'abord, ceci nécessite de disposer d'un algorithme performant d'estimation de mouvement. Puis, les vecteurs du mouvement calculés pourraient être utilisés pour obtenir une segmentation de l'image; dans un premier temps, les blocs de mouvement identique peuvent être reliés, et pour affiner, l'information des contours peut servir. Avec ces éléments, le traitement d'objets multiples serait tout à fait envisageable, un fond en mouvement ne serait plus problématique, la pollution par le bruit serait tout simplement annulée, et même, en incorporant de l'information temporelle ("extra shape information"), la définition d'un critère permettrait aussi bien la fusion que la fission des objets. Un avantage majeur serait la diminution de bits des vecteurs déplacement à transmettre, puisqu'un simple mouvement affine définirait le mouvement de chaque objet. Cependant, dans l'état actuel de leurs recherches, les auteurs restent sceptiques sur ce dernier point car leurs résultats finaux ne témoignent que d'une faible amélioration de la qualité visuelle et du rapport signal sur bruit.

Pourtant, ces remarques judicieuses sont décrites et appliquées dans des articles résumés dans ce qui suit. Chacun traite de la notion de codage video orienté-objet employant des champs denses pour la représentation du mouvement.

## 2. *Codage vidéo par segmentation des vecteurs de mouvement et des différences de trame.*

L'article [7] décrit un codage vidéo basé sur la double segmentation des vecteurs de mouvement et des différences entre trame. Afin de réduire le temps d'exécution, un détecteur de changement à deux seuils est introduit pour trouver les régions de l'image à mouvement notable. Alors, la procédure d'estimation des vecteurs de mouvement consiste en une détection des vecteurs (algorithme de comparaison de blocs  $2 \times 2$  appliqués aux régions obtenues par le détecteur de changement) suivie de leurs corrections sur les frontières, ainsi qu'à l'intérieur

des objets. Ensuite, une double segmentation est effectuée: segmentation des vecteurs mouvement et segmentation de la différence de trames pour les régions à grande erreur de prédiction (définie par différence entre l'image originale et l'image compensée-mouvement reconstruite par ces vecteurs). De plus, le signal vidéo de la différence de trame est utilisé, en guise de complément des résultats de segmentation. Le codage des contours-textures de l'image, une des techniques de codage d'images de seconde-génération considérant les propriétés visuelles humaines, s'appuie sur les segmentations préalablement réalisées. Cette méthode réduit l'effet de traîne, améliore le PSNR (Peak Signal to Noise Ratio) et le taux de compression.

### 3. *Codage vidéo orienté-objet.*

Dans les articles [8][9], les auteurs présentent une nouvelle approche "orienté-objet" par compensation de mouvement, dans le contexte du codage vidéo à très bas débit. En plus du codeur hybride conventionnel, ils ajoutent un estimateur et un codeur de champs denses de mouvement. La segmentation transforme le champ des vecteurs mouvement en régions de mouvement continu. Ainsi, les discontinuités du champ des vecteurs mouvement peuvent seulement apparaître aux frontières de régions. Un estimateur basé sur le critère MAP ("Maximum-A-Posteriori probability"), correspondant à une fonction objective, est employé dans l'estimation du champ de mouvement: il maximise la probabilité "a posteriori" du champ de déplacement d'une trame, étant données les images courante et précédente ainsi que les estimations du champ de déplacement des trames précédentes. Il prend aussi compte des relations temporelles statistiques le long des trajectoires de mouvement. De plus, un modèle théorique de séquences d'images est défini pour l'estimation des déplacements, et une procédure d'optimisation à plusieurs échelles de cette fonction objective est formulée. Chaque contour est approximé par un ensemble de fonctions polyno-

miales et splines. Comme la segmentation peut être prédite par compensation de mouvement de la trame précédente, seulement quelques vertex doivent être transmis de trame en trame. Les résultats donnent une bonne estimation du mouvement, cependant, la segmentation contient encore trop d'objets.

#### 4. *Codage vidéo 2-D basé-région pour le codage vidéo à très bas débit.*

Il est clair que le codage “basé-région” est étroitement lié au codage “basé-objet”. Nous allons introduire ici des éléments nouveaux sur la segmentation et l'estimation de mouvement qui devraient clarifier notre vue d'ensemble du codage vidéo à très bas débit.

Une méthode, découpant une trame en régions de forme arbitraire et de mouvement physiquement homogène, tout en respectant les trames précédentes de la séquence, est proposée dans [10]. Surtout, cette technique est spécialement conçue à des fins de codage et ne fait aucune supposition sur le contenu de la scène. Une modélisation du champ de déplacement ainsi qu'une segmentation sont conjointement réalisées et interagissent l'une avec l'autre. Plus précisément, l'identification du modèle de mouvement s'appuie sur une prédiction multi-échelle incluant la méthode de Gauss-Newton; au début de l'estimation, la région est à son état de plus faible résolution. A partir d'un modèle nul, quelques itérations sont effectuées. Le modèle résultant est alors rehaussé à une plus fine résolution jusqu'à la résolution finale. La segmentation du mouvement, pouvant être considérée comme un problème d'optimisation, utilise un processus hiérarchique de “split-and-merge” basé sur un arbre quaternaire, avec une étape additionnelle de réallocation, basée sur des relations de superposition définie sur l'arbre quaternaire: pour chaque niveau, les régions sont constituées de blocs connectés et ayant un même modèle de mouvement, réunis par fusion au travers des composantes connectées. Le passage d'un niveau à un autre permet l'affinement des régions. La procédure de fusion,



permettant la jonction des blocs de même mouvement, s'appuie sur l'étude d'un graphe pondéré décrivant l'adjacence entre régions où la pondération entre deux régions est définie par l'erreur moyenne de prédiction au carré (MSPE). Finalement, le codage de l'information du mouvement est fait par une représentation polynomiale du mouvement, des courbes de type B-splines codant les frontières des régions. Selon l'auteur, les résultats confirment l'excellente efficacité de la méthode, adaptée aux séquences arbitraires. De plus, le rapport qualité-débit est très satisfaisant.

Une considération-clé du standard MPEG-4 est la nécessité d'un compromis entre la qualité vidéo, surtout mesurée par le rendement spatio-temporel, et le débit vidéo. Les exigences supplémentaires sont le délai toléré au codeur et au décodeur, la performance du décodeur en présence d'erreurs de transmission et le coût de l'implantation. La série des méthodes décrites présente des solutions concrètes aux inconvénients introduits par le très bas débit. Enfin, les dernières techniques algorithmiques constituent un apport considérable au codage vidéo à très bas débit, permettant une meilleure compression vidéo: c'est le cas du codage vidéo 2-D "basé-région".

### **2.3. Intérêt de recherche à l'INRS sur le codage vidéo à très bas débit**

Le groupe des Communications Visuelles de l'INRS-Télécommunications est impliqué dans une recherche portant sur ce sujet. L'activité de recherche est constituée de six blocs schématisés à la Fig. 2.2: segmentation intra-trame, segmentation et estimation du mouvement inter-trame, codage intra-trame, codage inter-trame, codage des cartes de segmentation, codage des paramètres de mouvement. Ce découpage a été envisagé pour les raisons suivantes.

La représentation numérique des images requiert une grande quantité de bits. Le but principal du codage est de réduire cette quantité et de reconstruire à l'aide du décodeur une copie conforme à l'image originale. Faisant suite à un ensemble de techniques de première-génération, les études récentes ont prouvé que le système visuel humain (SVH) est sensible à certaines propriétés d'image. C'est pourquoi des méthodes de codage de seconde-génération utilisent de nouvelles approches [11][12], en tentant de décrire une image en terme de contours et textures. Ces éléments semblent plus logiques puisqu'ils coïncident avec certains concepts physiologiques de la vision, mais surtout, la propriété de stationnarité est conservée dans toute région. De plus, l'effet de masque, intervenant aux frontières des régions, peut être pris en compte. Plus précisément, chaque région d'une scène naturelle est caractérisée par sa frontière (contours) délimitant les objets (texture). Ceci signifie que l'extraction des contours d'une image segmentée est de première importance; rappelons que la segmentation, dont une explication détaillée sera donnée plus loin, consiste à diviser, selon un modèle, l'image en une classe de régions contiguës de forme arbitraire, si bien que le résultat final ressemble à un puzzle. Généralement, l'information résultante à coder  $R_i$ , pour l'objet  $i$ , possède trois composantes de débit  $R_{Ci}$ ,  $R_{Ti}$ ,  $R_{Mi}$  représentant respectivement l'information des contours, de la texture et du mouvement. Comme un compromis doit être établi entre ces trois débits [13], une priorité de contrôle dans les nouveaux codeurs décide quel type d'information doit être transmis pour chaque objet. Alors que le taux de compression atteignait avec peine le plateau des 10:1 il y a vingt ans, ces nouveaux critères pourraient permettre une compression allant jusqu'à 200:1.

Dans la description qui va suivre, nous allons plus particulièrement nous intéresser aux blocs de segmentation d'images, de représentation des textures et de codage des

cartes de segmentation.

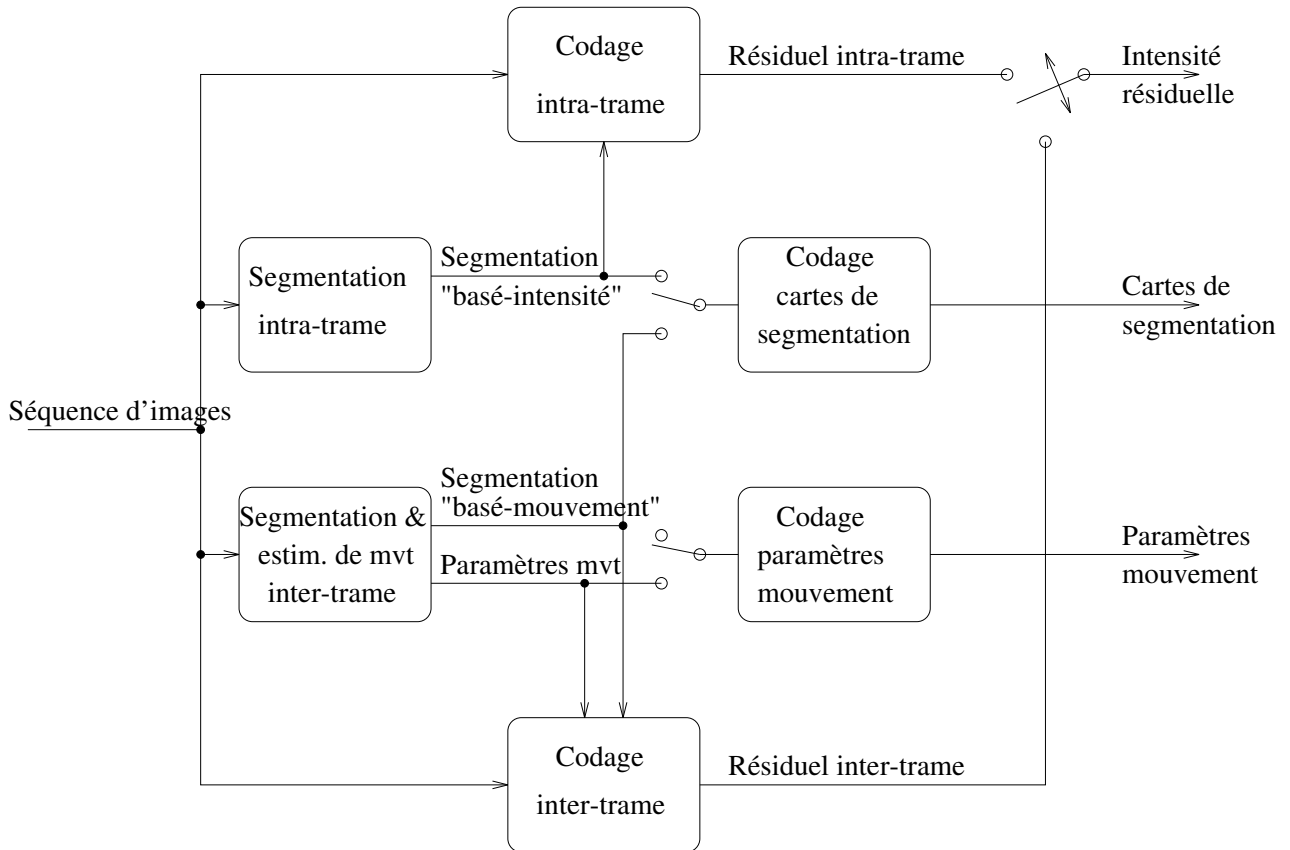


FIG. 2.2 - Exemple de diagramme d'un système de codage vidéo "basé-région" pour des applications à très bas débit

### 2.3.1. Segmentation de l'image

Le processus de segmentation d'image est primordial pour le projet, car tout ce qui sera entrepris à son sujet aura un énorme impact sur la nature des résultats de compression des cartes de segmentation. Dans le groupe des Communications Visuelles de l'INRS-Télécommunications, les algorithmes de segmentation d'image sont encore en cours de mise au point. Deux types de segmentations ont été dévelop-

pées:

1. *Segmentation intra-trame basée-luminance.*

Il existe une méthode de segmentation bien connue, appelé “split-and-merge”, mentionnée au point 4 de la section 2.2.5. Elle consiste à partitionner l’image en blocs carrés, puis à effectuer une fusion-fission de ces blocs basée sur un critère d’homogénéité donné. Ce critère est souvent choisi égal à la variance de la répartition de l’intensité sur les blocs. Si la variance est supérieure à un seuil, le bloc est cassé en quatre sous-blocs, auxquels le processus de fission est récursivement appliqué. Par contre, deux blocs voisins peuvent être fusionnés en fonction de la variance. La double procédure fusion-fission est itérative.

Cependant, nous lui avons préféré une autre technique de segmentation utilisant l’algorithme développé dans [14] et qui calcule une estimation de la longueur de description minimale (MDL), conditionnée sur l’image originale. Soit  $D^n$  la description de l’image correspondant à la  $n$ -ième image  $I^n$  de la séquence, et  $L$  la fonction de longueur de code idéale. La description optimale  $D^n$  conditionnée sur  $I^n$  minimise:

$$L(D^n/I^n) = L(I^n/D^n) + L(D^n), \quad (2.1)$$

Dans le cas de segmentation, la description  $D^n$  est modélisée en une image constante par parties, et l’image  $I^n$  comme la somme  $I^n = D^n + \eta$ , où  $\eta$  est un bruit blanc stationnaire gaussien de moyenne nulle et de variance  $\sigma^2$ . Le premier terme de droite dans l’équation MDL définit le codage d’entropie de la variable aléatoire  $\eta = I^n - D^n$ , tandis que le second terme est la mesure de la complexité associée à la description  $D^n$ . Cette mesure de complexité est choisie comme le nombre de points frontières dans  $D^n$ . De ce fait, l’équation

peut être reformulée par:

$$L(D^n/I^n) = 1/2 \sum_i (D_i^n - I_i^n)^2 / (2\sigma^2) + 1/2 \log(2\pi\sigma^2) \\ + 1/2 \sum_i \sum_{j \in N(i)} (1 - \delta(D_i^n - D_j^n))$$

expression dans laquelle  $(i, j)$  sont les indices des pixels de l'image,  $N(i)$  un voisinage du pixel  $i$  et  $\delta()$  symbolise la fonction de Dirac. On retrouve dans cette équation une similitude frappante aux formulations bayésiennes d'un problème de segmentation. Par analogie, le calcul de l'estimation MDL s'effectue par procédure itérative de type recuit déterministe ("deterministic annealing"). Malheureusement, la technique de recuit déterministe peut converger très lentement vers l'estimation. Afin d'apporter une compensation à cet inconvénient, le processus est interrompu après un certain nombre d'itérations (quand l'estimation est assez proche de la valeur optimale), et est suivi par une procédure de regroupement ("clustering") basée-luminance. La dernière étape consiste à étiqueter l'image pour l'obtention de la segmentation désirée. Un exemple est donné à la Fig. 3.1.

## 2. Estimation du mouvement et segmentation jointes "basé-région".

Le but de l'estimation du mouvement et de la segmentation jointes est de partitionner l'image en régions de mouvement homogène. Dans le contexte du codage vidéo, une réduction du débit de transmission peut être obtenue en utilisant des modèles paramétriques (de deux à douze paramètres) pour la description du mouvement des pixels d'une région [15]. En général, les méthodes d'estimation-segmentation du mouvement suivent l'une des deux approches ci-après:

- dans la première, à partir de la différence de trames comme segmentation initiale, les paramètres de mouvement seront estimés selon un modèle qui

minimise l'erreur de prédiction. Les pixels, qui ne vérifient pas le modèle seront regroupés pour former de nouvelles régions, ce processus étant répété jusqu'à convergence [16].

- La seconde méthode est basée sur le critère MAP pour établir l'interaction entre l'estimation et la segmentation du mouvement [9]. En pratique, ceci revient à faire l'optimisation d'une fonction d'énergie, composée de quelques termes modélisant l'erreur de prédiction compensée par le mouvement, ainsi que de certaines hypothèses sur la forme à priori des régions.

### **2.3.2. Représentation de la luminance: codage par transformée des textures de régions de forme arbitraire**

Ici aussi, deux méthodes sont à l'étude:

- dans la première, on utilise les règles d'un codage par transformée, en adaptant une transformée DCT sur des régions de forme arbitraire d'une image [17] [18] [19]. Pour cela, nous considérons une base de vecteurs associée à la transformée sur un carré circonscrit à l'objet considéré et une orthogonalisation s'opère selon la forme de l'objet. Deux procédés d'orthogonalisation ont été proposés; leur structure permet de respecter au mieux les propriétés de la base originelle (propriété de symétrie et de distribution fréquentielle). L'évaluation de la méthode est basée sur le critère en concentration d'énergie. En ce qui concerne la représentation de la texture, les résultats sont très satisfaisants. Toutefois ce procédé nécessite beaucoup d'espace mémoire pour traiter la texture de chaque région d'une image [20],
- dans la seconde méthode, les transformées classiques sont encore appliquées sur des régions de formes arbitraires, mais inscrites dans le plus petit rectangle. Après transformée FFT (Fast Fourier Transform) de ce bloc, le spectre obtenu

peut être considéré comme la convolution du spectre  $S$ , associé à la texture de la région, et de celui associé au signal de la fenêtre carrée contenant la région. Afin de calculer  $S$ , un processus itératif est lancé, et arrêté quand l'erreur moyenne quadratique (MSE) est inférieure à un seuil prédéterminé. Ensuite, les coefficients du spectre  $S$  sont estimés par une étape itérative décrite dans [21]. Ce travail est encourageant, mais il reste aussi extrêmement coûteux.

### 2.3.3. Codage des cartes de segmentation

En Fig. 2.3, le diagramme du codage d'image "basé-région" schématise les deux types d'images sources à coder. Il s'agit des deux types d'images segmentées décrites plus haut: intra- et inter-trame. Nous avons vu que certaines descriptions de codage basé-région pour le très bas débit mentionnaient quelques solutions reliées au codage de l'information de segmentation. Par exemple, dans [8], [9] et [10], les auteurs ont décidé d'approximer les contours des régions de l'image par des fonctions B-splines ou des fonctions polynômes. Bien que cette information de segmentation représente une grosse part des données globales de l'image à coder, on dispose encore de peu d'informations démontrant l'efficacité d'une technique plutôt qu'une autre. En d'autres termes, peu d'études comparatives ont été menées jusqu'à présent sur le problème du codage des cartes de segmentation. C'est pourquoi nous consacrerons les deux prochains chapitres à une étude comparative de diverses techniques de codage sans perte et avec perte des cartes de segmentation.

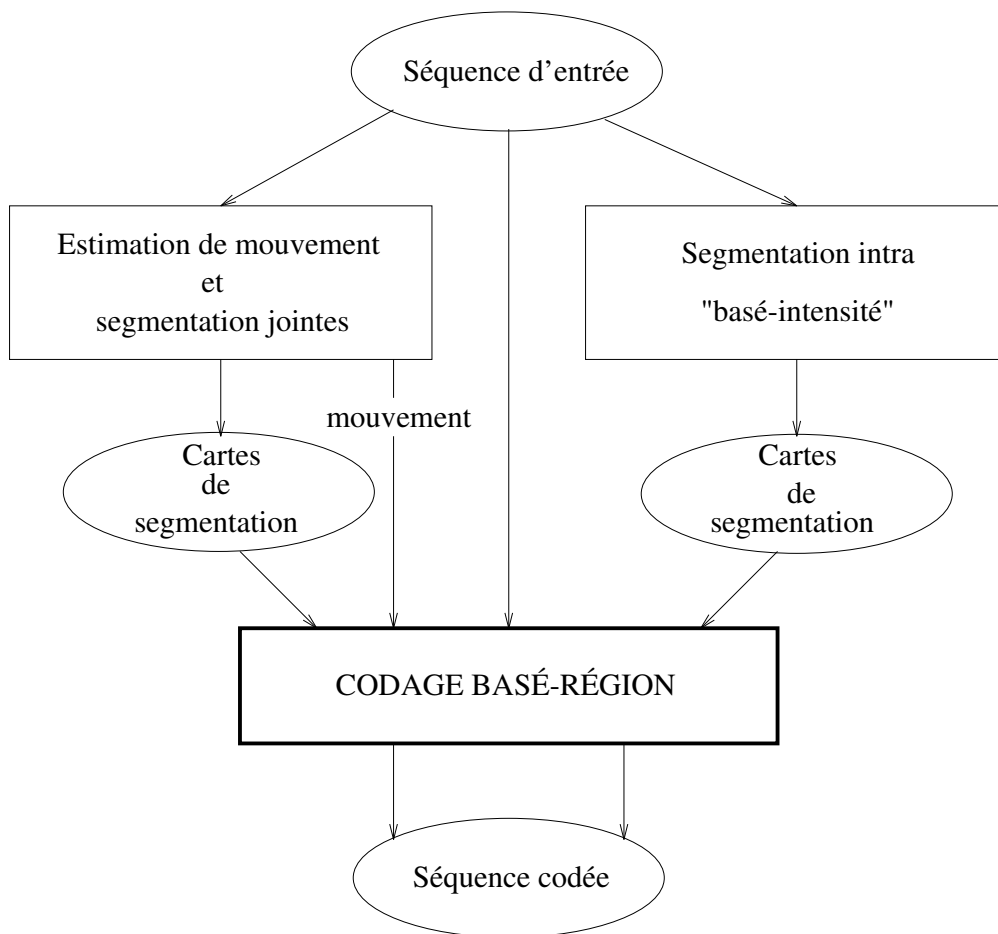


FIG. 2.3 - Codage vidéo à très bas débit; rôle du codage des cartes de segmentation



# Chapitre 3

## Codage sans perte

Ce chapitre se concentre sur l'étude de codages sans perte des cartes de segmentation. Pour cela, certains algorithmes sont décrits ainsi que les modifications personnelles apportées. Puis, l'intérêt des diverses techniques présentées est commenté et jugé, au travers de premiers résultats. Dans ce chapitre, seulement ceux impliquant la segmentation intra-trame sont cités. Le chapitre 5 fera état de l'ensemble des résultats pour les deux segmentations intra- et inter-trame.

### **3.1. Description globale du codage des cartes de segmentation**

Les cartes de segmentation sont définies par un ensemble d'étiquettes, chaque étiquette correspondant à une région d'intensité uniforme, obtenue par la procédure de segmentation (Fig. 3.1). Toujours d'après les trois exemples de la Fig. 3.1, chaque carte de segmentation possède un certain nombre d'états égal au nombre d'étiquettes, qui représentent les différentes régions. Or, la décorrélation de ces états est généralement loin d'être assurée. C'est pourquoi, au lieu d'utiliser les techniques usuelles de codage (telles que DPCM, codage prédictif ou bien quantification vectorielle) qui,

théoriquement, ne sont pas adaptées pour le codage de ce type d'images, nous allons nous pencher sur le synoptique de la Fig. 3.2 qui symbolise les étapes du codage de l'information de segmentation; l'étape de décorrélation des états va être prise en charge par la détection des contours de l'image segmentée. Ensuite, un algorithme efficace de codage de contours (idéalement, de l'information décorrélée) doit être mis en place pour donner un bon taux de compression. Enfin, la partie décodage traite de l'opération inverse. En d'autres termes, le travail entrepris est détaillé dans les points 1 et 2 suivants.

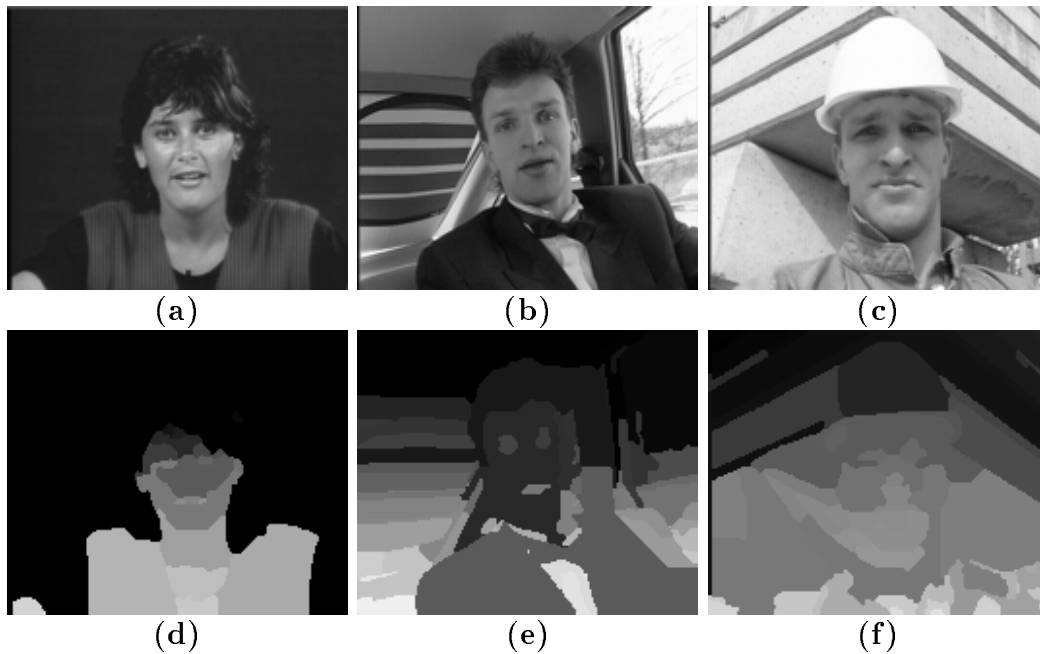


FIG. 3.1 - Trame #0 des séquences (a) “Miss America”, (b) “Carphone”, (c) “Foreman” et un exemple de segmentations pour (d) “Miss America” (21 régions, PCM: 127 kbits), (e) “Carphone” (74 régions, PCM: 177 kbits), (f) “Foreman” (91 régions, PCM: 177 kbits) obtenues par la méthode basée sur MDL

1. *Extraction et codage des contours de l'image segmentée.*

L'information des contours peut être élaborée à partir de deux procédés. En premier lieu, en appliquant un balayage spécial, les transitions horizontales sont extraites (Fig 3.15), permettant de transformer les cartes  $R$ -aires (avec  $R$

le nombre d'étiquettes) en une source 4-aire (cf Fig. 3.3). Cette source 4-aire peut ensuite être codée par des méthodes impliquant n'importe quel type de codage binaire sans perte, tel que les codages RLC ou BAC. Pour un codage sans perte, cette représentation doit coder assez d'information pour assurer que le processus de décodage-reconstruction rétablit la copie conforme de la séquence vidéo originale. En second lieu, les contours entiers des régions sont extraits et les chaînes ainsi obtenues peuvent être directement codées, soit par un codage sans perte (ex: CC), soit par un codage avec perte effectuant une approximation polygonale optimale des contours de régions.

## 2. *Décodage et reconstruction de l'image segmentée.*

En supposant qu'aucune erreur de transmission n'intervienne dans le canal, la reconstruction doit être parfaite pour le codage sans perte. Dans le cas des techniques sans perte impliquant le codage des transitions horizontales, nous offrons un algorithme de reconstruction basé sur le phénomène de cohérence d'étiquettes. Pour le cas du codage de chaînes, nous allons utiliser un algorithme de remplissage d'intérieurs de contours. Enfin, la reconstruction par codage avec perte consistera à remplir l'intérieur des polygones approximatés, définissant la nouvelle carte de segmentation.

## 3.2. Codage des transitions horizontales

En extrayant les états de transition horizontale d'une image segmentée, nous obtenons une image binaire  $\{N, B\}$ ;  $B$  correspondant par exemple à une transition et  $N$  au cas contraire. A partir de cet instant, n'importe quel codage d'images binaires peut être appliqué pour obtenir la compression désirée. Or, il existe depuis bien longtemps toute une hiérarchie de techniques de compression pour les images binaires, citées dans [22]. Les plus connues sont celles qui ont été développées pour

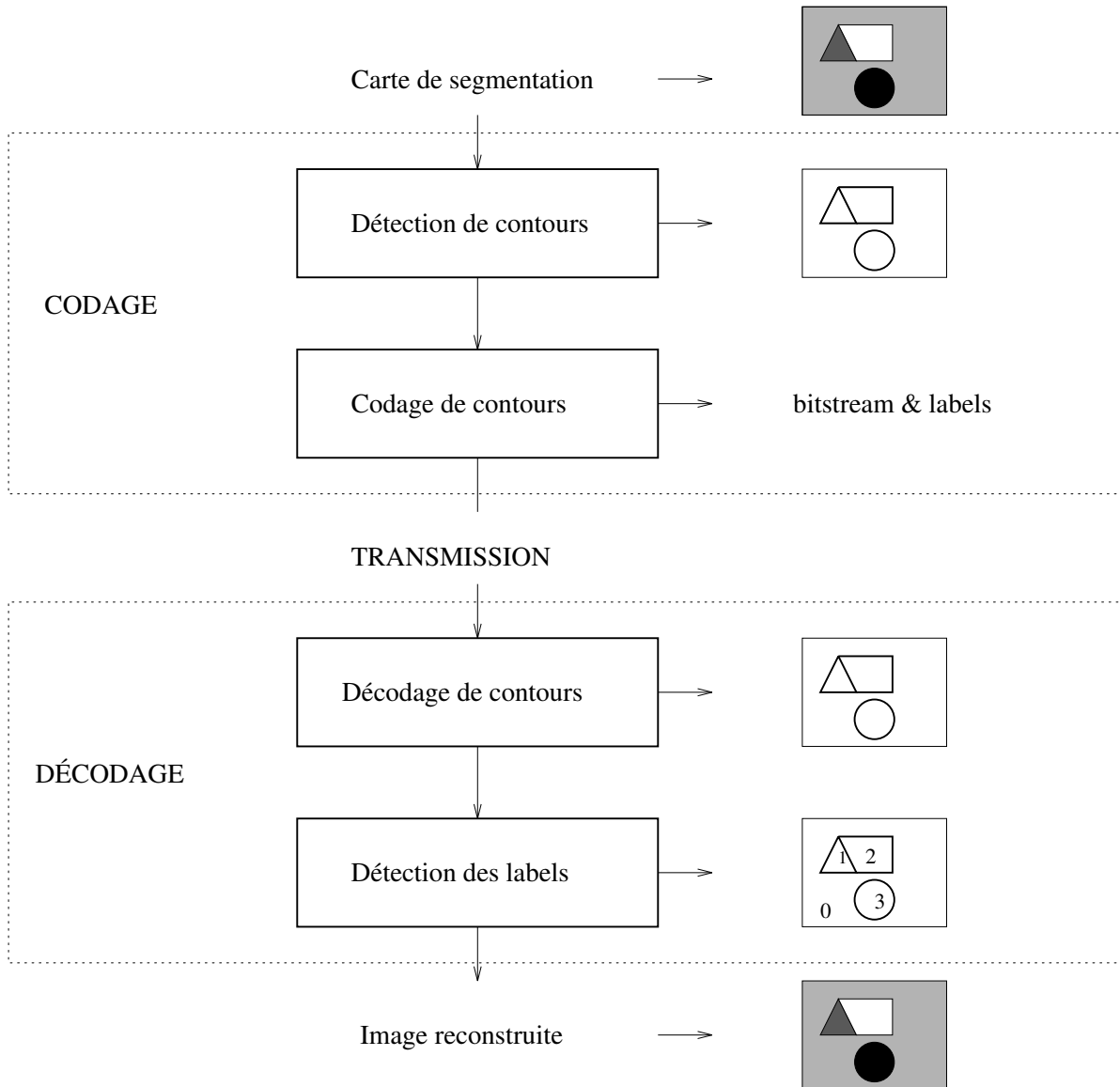


FIG. 3.2 - Codage/décodage des cartes de segmentation

la technologie du facsimile et se raccordent aux standard G.3 et G.4.

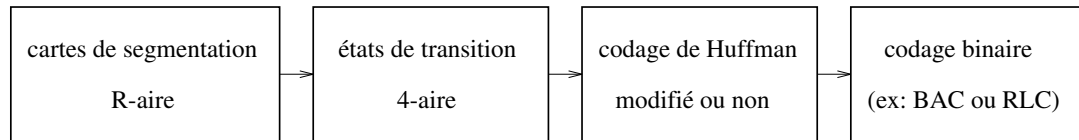


FIG. 3.3 - Principe de codage des cartes de segmentation, en utilisant les états de transition horizontale

### 3.2.1. Extraction des états de transition de cartes de segmentation

La Fig. 3.5 donne un exemple simple d'une carte de segmentation, constituée de trois objets sur un fond; ainsi, chaque pixel peut prendre seulement quatre valeurs d'étiquette. L'image des états de transition est définie sur la même grille que l'image segmentée. D'abord, nous effectuons un balayage spécial ligne par ligne, de façon à préserver la continuité verticale à la fin de chaque ligne balayée. Puis nous marquons, par certains codes, les transitions horizontales, dès que nous les rencontrons; il y a une transition horizontale à un pixel si sa valeur diffère de celle du pixel précédent selon l'ordre de balayage. Remarquons que les cases vides correspondent au code 0. Le codage est construit pour prendre en considération la volonté d'un décodage efficace. Aussi, dans le but d'utiliser un patron simple pour le processus de reconstruction, une seule hypothèse doit être prise en compte dans l'étape de segmentation: elle doit interdire aux régions des longueurs de transition horizontale de 1 ou 2 pixels.

Nous allons rencontrer deux types de pixels à coder. La première famille est formée des propres pixels de transition horizontale. Et pourtant, comme il nous faut aussi coder l'information de transition verticale, nous allons utiliser des mots-codes en nombre restreint (ici notées de 0 à 3) pour coder ces situations. Soulignons que

cette représentation à quatre états est originale à l'auteur. Le processus d'attribution des quatre mots-codes dépend énormément du processus de reconstruction: les règles dans l'étape de reconstruction doivent être connues pour l'attribution des mots-codes. C'est pourquoi, chaque fois que l'on attribue un code au pixel courant, la ligne précédente dans la direction courante du balayage est analysée. La longueur du patron est ainsi variable et peut s'étendre de trois pixels à toute la ligne précédente. Décrivons l'attribution des quatre mots-codes:

**code 0** : il est utilisé dans le cas de non transition horizontale, ce qui signifie que nous sommes à l'intérieur d'une région et, de ce fait, rien de bien particulier doit être marqué.

**code 1** : il est réservé pour de "simples" transitions; en fait, nous avons déjà rencontré pendant le balayage l'étiquette du pixel courant. En conséquence, nous avons juste besoin de coder la transition, et non l'étiquette en elle-même, car elle est présente sur la ligne précédente dans le sens du balayage courant. Pour attribuer le mot-code 1, nous utilisons un patron adaptatif: il sera étiré, sur la ligne précédente dans la direction courante du balayage, jusqu'à rencontrer l'étiquette du pixel courant.

**code 2** : il est utilisé dans deux cas,

- nous rencontrons pour la première fois l'étiquette du pixel courant, ce qui signifie que nous rentrons dans une nouvelle région; ou bien, l'étiquette du pixel courant n'est pas disponible à la ligne précédente, dans la direction courante du balayage (Fig. 3.4(a)). Pour ces deux situations, le mot-code 2 signifie que l'on attribue une nouvelle étiquette au pixel courant, car dans l'étape de reconstruction, on fera face à un cas d'indétermination.
- nous associons le couple de codes  $\langle 2 - 2 \rangle$ , i.e. deux codes 2 espacés que par les codes 0 (ex: 200000...0002) à certaines arêtes horizontales

(i.e. transitions verticales) au bas de régions (Fig. 3.4(b)); ces arêtes horizontales correspondent aux frontières horizontales entre plusieurs régions de même niveau et une autre région (dans le cas de la Fig. 3.4(b), il s'agit de la frontière entre les régions A-B et C). La nécessité de faire l'usage de ce type de codage (200000...0002) provient de la méthode choisie lors du processus de reconstruction: une analyse par paires de mots-codes est effectuée pour allouer la bonne étiquette à chaque pixel.

Il est utile de signaler qu'il ne peut y avoir de confusion dans l'utilisation de ce même couple  $\langle 2 - 2 \rangle$  pour les cas distincts décrits. En effet, lors du décodage, nous vérifions que le couple de codes  $\langle 2 - 2 \rangle$  correspond, oui ou non, à une transition verticale attendue.

**code 3** : il sert à coder des "coins" de régions, que nous ne pouvions coder avant, et qui s'avèrent indispensables pour la reconstruction; dans le patron choisi pour le processus de reconstruction (un nombre adaptatif de pixels situés au dessus du pixel courant sur la ligne précédente), l'étiquette du pixel courant n'est pas disponible. Le mot-code 3 joue alors le rôle d'indicateur de l'étiquette du pixel courant, en le positionnant sous le premier pixel (de la ligne précédente dans la direction courante du balayage) de même étiquette que le pixel courant.

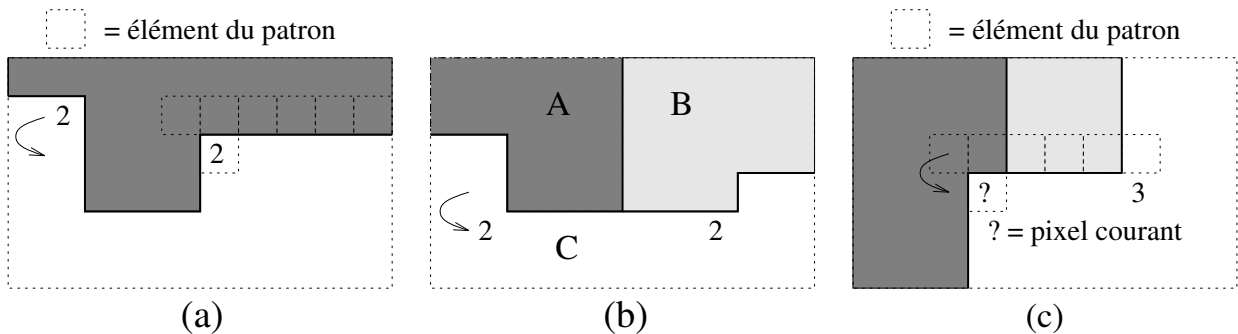


FIG. 3.4 - Exemples d'attribution des mots-codes 2 et 3

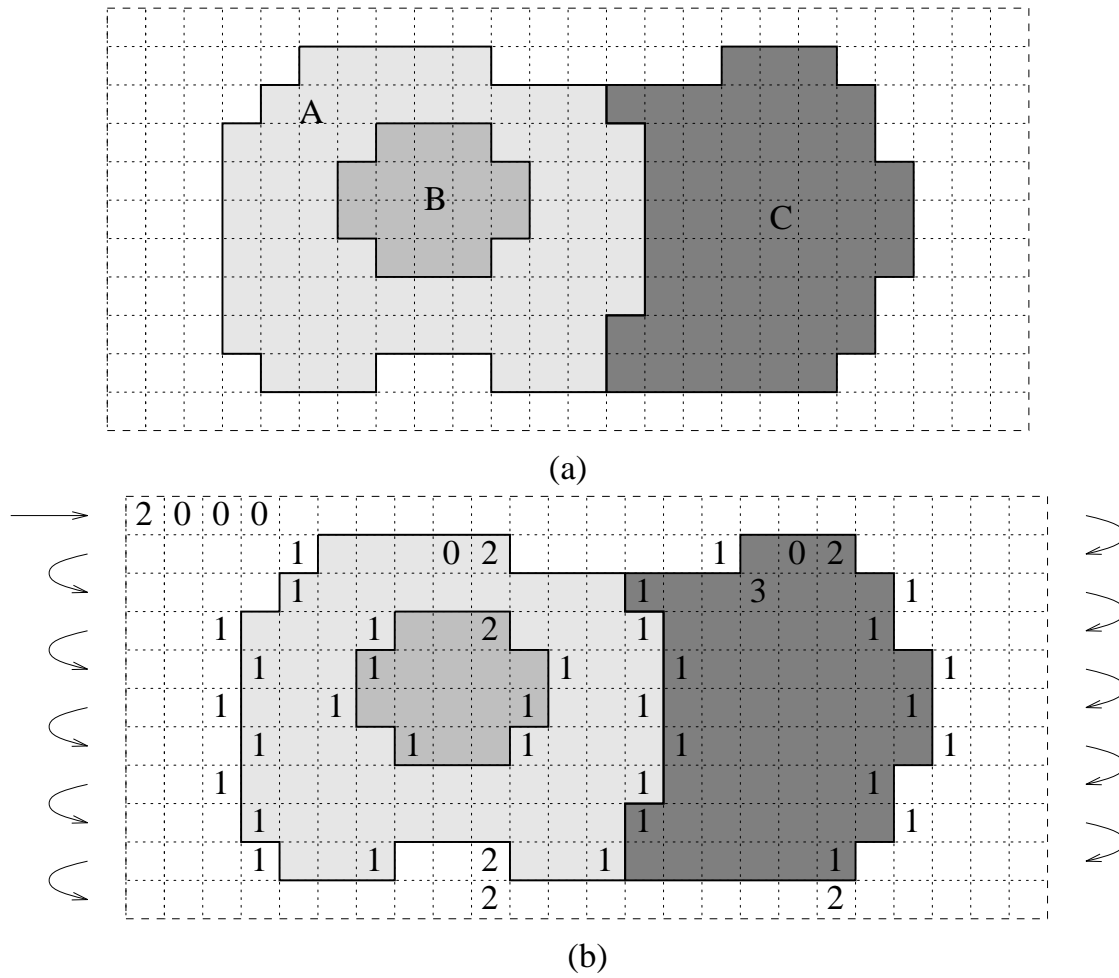


FIG. 3.5 - Exemple d'une (a) segmentation avec (b) différents états de transition



Pour conclure, les mots-codes 1 et 2 sont des marqueurs de transitions, alors que le mot-code 3 est un indicateur d'étiquettes. Dans l'exemple de la Fig. 3.5, une partie du flux des états de transition devient:

```
200000000000000000000000000000020010000020  
00010000000010000000010030001000000010...
```

### 3.2.2. Processus de reconstruction

Le processus de reconstruction s'inspire d'une méthode basée sur la cohérence des étiquettes ("labelling consistency"). Plusieurs notions sont exposées dans l'article [23], présentant un processus général sur l'étiquetage par relaxation continue. L'idée directrice consiste à assigner, pendant le décodage, une nouvelle étiquette à un pixel chaque fois que l'on rencontre une indétermination, puis de lever les incohérences en fusionnant des régions à partir des différentes étiquettes. Dans notre cas, le processus de cohérence est rendu possible par l'information des codes 2 et 3. Certaines situations types de processus de fusion sont schématisées en Fig. 3.6:

- (a) il s'agit du cas le plus courant. Une simple transition a été marquée par le code 1. Il n'y a aucun problème majeur de reconstruction; il suffit d'appliquer un patron (indiqué par les cases en pointillé sur la Fig. 3.6 (a)), à la ligne précédente, pour rechercher l'étiquette du pixel courant qui, notamment, a déjà été rencontrée à la ligne du dessus,
- (b) un cas identique à (a) est décrit: le coin de la région n'a pas été marqué car la reconstruction est simple; comme nous savons que l'étiquette du pixel a déjà été rencontrée (car le code 1 a été utilisé), il suffit encore de la rechercher sur la ligne précédente à l'aide d'un patron de longueur adaptative,
- (c) le coin de la région a été marqué par le code 3, car on s'est rendu compte que l'étiquette du pixel courant n'était pas disponible dans le patron considéré pour

la reconstruction. Maintenant, l'indétermination est facilement levée puisque l'étiquette recherchée est la même que le pixel juste au dessus,

- (d) on a représenté les deux cas les plus fréquents de rencontre des mots-codes. En fait, il s'agit toujours de la présence des mots-codes 1, 2, ou 3 espacés par le code 0. Le but de la reconstruction est donc d'étudier chaque couple de mots-codes distincts de 0, puis de prendre une décision sur le patron adaptatif à employer pour l'étiquetage final.
- (e) ceci est un exemple de l'utilisation du code 2 pour marquer la rencontre d'une nouvelle étiquette (cas d'indétermination). Dans cet exemple, un processus de cohérence d'étiquetage suivra, pour joindre les parties gauche et droite d'une même région. Ce processus de cohérence s'effectue par l'étude des paires de mots-codes (ici, il s'agit de la paire  $\langle 1 - 1 \rangle$  correspondant à une arête de la région foncée),
- (f) nous avons codé une arête de région avec le couple  $\langle 2 - 2 \rangle$  disposé aux extrémités de l'arête. Ceci nous permet de faire la fusion de parties de régions déjà décodées, et de lever des cas d'incohérence préalablement insérés. Nous voyons encore dans cet exemple que nous sommes amenés à faire l'analyse des paires de mots-codes pour lever les indéterminations d'étiquettes.

D'une certaine mesure, ce processus de reconstruction étudie chaque couple d'états de transition, distincts du code 0. Ces couples définissent des frontières entre régions (coins, bordures verticales ou horizontales ...). Les couples rencontrés sont toutes les combinaisons possibles de binômes construits à l'aide des codes 1, 2 et 3. Nous forçons le lien de ces différents couples par le procédé de cohérence dans l'étiquetage. La reconstruction effectuée est sans perte.

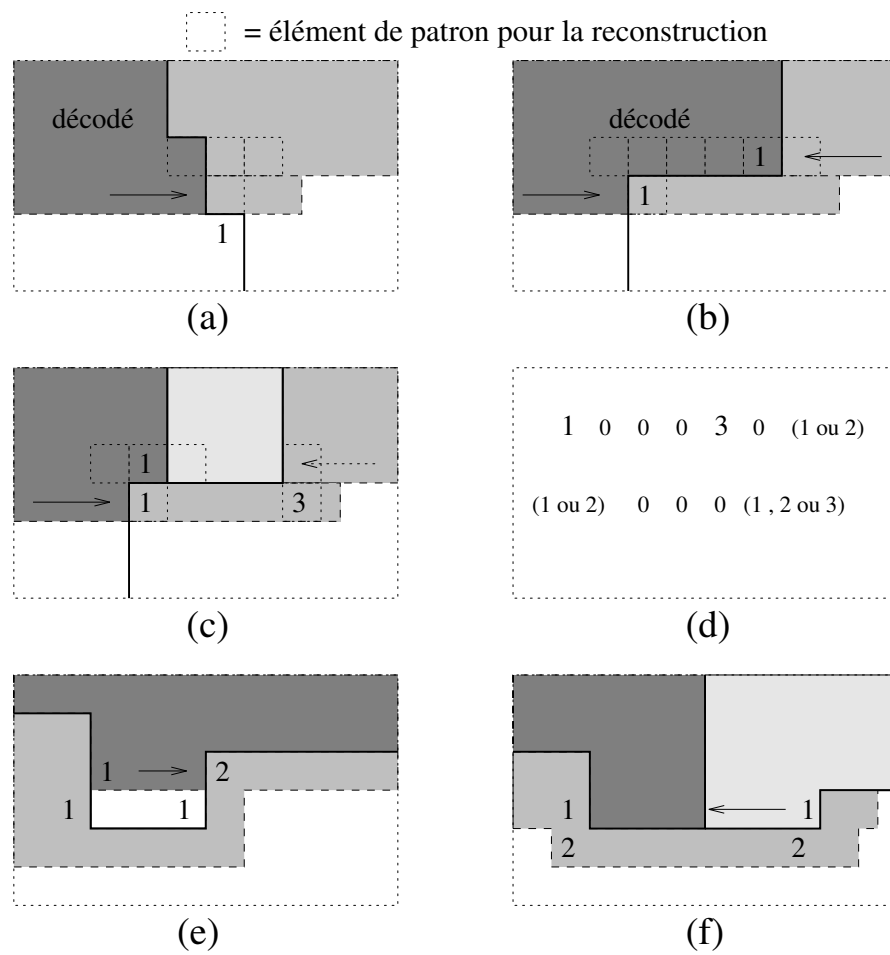


FIG. 3.6 - Processus de reconstruction basé sur la cohérence d'étiquetage

### 3.2.3. Codage effectif des états de transition

Nous devons coder le flux de bits représentatif des états de transition. Ce flux est constitué de quatre mots-codes. Pour cela, nous allons transformer ces quatre mots-codes en appliquant une table de Huffman [24]. Cette table de Huffman est construite à partir des histogrammes des quatre mots-codes, calculés sur trois images de référence (voir Tableau 3.2). Le Tableau 3.1 propose la construction de la table de Huffman pour le cas décrit en Fig. 3.5. Notons la similitude de ces deux Tableaux. Désormais, nous pouvons utiliser n'importe quel type de codage binaire (Fig. 3.3).

Code	Distrib.	Huffman
0	88%	0
1	9%	10
2	2%	110
3	1%	111

TAB. 3.1 - Table de Huffman associée à la Fig. 3.3 ( $\hat{H} = 0.65, \hat{L} = 1.15$ )

Dans ce chapitre, nous allons nous concentrer sur deux codages. Le premier est le codage binaire non adaptatif RLC. Le second met l'accent sur une technique adaptative: le codage arithmétique binaire BAC. Mais avant de rentrer plus dans le détail, définissons, pour une source  $\mathbf{S} = \{s_0, s_1, \dots, s_{N-1}\}$  de  $N$  symboles, l'entropie marginale d'ordre un  $H(\mathbf{S})$  et la longueur moyenne  $L$  du code par

$$H(\mathbf{S}) = \sum_{i=0}^{N-1} p_i \log_2\left(\frac{1}{p_i}\right)$$

$$L = \sum_{i=0}^{N-1} p_i l_i$$

avec, pour  $i = 0, \dots, N - 1$ ,  $p_i$  la probabilité et  $l_i$  la longueur du code du symbole  $s_i$ . Les estimations  $\hat{H}$  et  $\hat{L}$  sont exprimées en bits par symbole. Nous aurions aussi pu procéder de la manière suivante: appliquer une Table de Huffman pour deux symboles joints. Les histogrammes des trois séquences sont inscrits au Tableau 3.3.

Code	"Miss America"		"Carphone"		"Foreman"	
	Distrib.	Huff.	Distrib.	Huff.	Distrib.	Huff.
0	98%	0	95.6%	0	94%	0
1	1.75%	10	4%	10	5.5%	10
2	0.1%	110	0.3%	110	0.4%	110
3	0.05%	111	0.1%	111	0.1%	111

TAB. 3.2 - Tables de Huffman utilisées pour "Miss America" ( $\hat{H} = 0.15, \hat{L} = 1.02$ ), "Carphone" ( $\hat{H} = 0.28, \hat{L} = 1.05$ ) et "Foreman" ( $\hat{H} = 0.36, \hat{L} = 1.07$ ). Distributions évaluées sur les séquences segmentées

Le fait de joindre par deux les symboles nous permet de diminuer notre estimation  $\hat{L}$ . Nous pourrions aussi étendre cette méthode à 3, 4,  $\dots$ , symboles joints.

Code	"Miss America"		"Carphone"		"Foreman"	
	Distrib.	Huff.	Distrib.	Huff.	Distrib.	Huff.
00	96%	0	91.6%	0	87.7%	0
10	1.9%	10	4.4%	10	6.17%	10
01	1.77%	110	3.37%	110	5.1%	110
20	0.2%	1110	0.3%	1110	0.5%	1110
02	0.1%	11110	0.2%	11110	0.4%	11110
03	0.02%	111110	0.07%	111110	0.07%	111110
30	0.01%	111111	0.06%	111111	0.06%	111111
<i>autre</i>	0%	-	0%	-	0%	-

TAB. 3.3 - Tables de Huffman en codant conjointement deux symboles, pour "Miss America" ( $\hat{H} = 0.3, \hat{L} = 1.07$ ), "Carphone" ( $\hat{H} = 0.54, \hat{L} = 1.14$ ) et "Foreman" ( $\hat{H} = 0.72, \hat{L} = 1.20$ ).  $\hat{H}$  et  $\hat{L}$  sont ici exprimés en bits par deux symboles joints. Les distributions ont aussi été évaluées sur les séquences segmentées

### 3.2.4. Codage de la longueur des courses (RLC)

La description du RLC fait référence à l'article [25]. Nous considérons le cas d'un événement binaire de probabilités  $p$  et  $q$ , avec  $p \gg q$  et  $q = 1 - p$ ;  $p$  et  $q$  coïncident respectivement avec les événements favorable et non favorable. Le Tableau 3.2 prouve clairement que le bit 0 du flux est bien plus probable que le bit 1. L'idée est de coder la longueur entre les apparitions successives de l'événement le moins probable (ici

bit 1). Couramment, la probabilité de telles courses de taille  $n$  est  $p^n q$ , qui définit une distribution géométrique. Si la liste de ces courses était finie, nous pourrions les associer à leurs probabilités et appliquer un codage de Huffman. Or, avec une liste infinie, nous devons faire certaines hypothèses.

Soit  $p^m = \frac{1}{2}$ , où  $m$  est entier. Dès lors, une course de taille  $n + m$  est moitié moins probable que la course de taille  $n$ . On s'attend à ce que le mot-code, associé à la course de taille  $n + m$ , ait un bit de plus que pour celui de taille  $n$ . Il y aura donc  $m$  mots-codes pour chaque longueur de code possible, excepté les longueurs de mots les plus courts. La construction de dictionnaires (ex: Tableau 3.4), en fonction de la valeur  $m$ , s'appuie sur les remarques précédentes. Enfin, soit  $k$  le plus petit entier positif tel que  $2^k \geq 2m$ . Alors, le dictionnaire de mots-codes contient exactement  $n$  mots de longueur  $\geq k$ , ainsi que  $2^{k-1} - m$  mots de longueur  $k - 1$ .

Il n'y a pas de problème particulier pour le décodage. Nous allons considérer  $m = 16$ , ce qui nous donne  $p \simeq 0.95$ . Les règles de décodage sont les suivantes: en partant de la gauche du mot, nous comptons le nombre de 1 précédant le premier 0. Soit  $A \geq 0$  ce nombre. Alors le mot a  $A + 5$  bits. Les 5 derniers bits sont la représentation ordinaire binaire de l'entier  $R$ ,  $0 \leq R \leq 15$ ; le décodage correct du mot est finalement  $16A + R$ .

Un problème d'efficacité de codage pourrait provenir de la valeur supposée  $p$ . Comme nous considérons une image binaire où seuls les états de transition sont signalés, cette valeur  $p$  est proche de l'unité. D'après les histogrammes, le choix  $m = 16$ , donnant  $p \simeq 0.958$  est acceptable.

#### 3.2.4.1. Conclusion

$m = 16$							
$n$	mot-code	$n$	mot-code	$n$	mot-code	$n$	mot-code
0	00000	8	01000	16	100000	24	101000
1	00001	9	01001	17	100001	25	101001
2	00010	10	01010	18	100010	26	101010
3	00011	11	01011	19	100011	27	101011
4	00100	12	01100	20	100100	28	101100
5	00101	13	01101	21	100101	29	101101
6	00110	14	01110	22	100110	30	101110
7	00111	15	01111	23	100111	31	101111

TAB. 3.4 - Partie de dictionnaire du RLC pour  $m = 16$ 

	bits	bit/pixel
“Miss America”	4053	0.160
“Carphone”	8016	0.316
“Foreman”	9327	0.368

TAB. 3.5 - Nombre de bits, pour la trame #0 segmentée de chaque séquence, par codage RLC

Le Tableau 3.5 propose un premier résultat de compression à l’aide du codage RLC. La compression est nettement meilleure que pour un codage classique PCM. En étant toutefois conscient des buts fixés par le codage à très bas débit ( $\simeq 16$  kbits/s avec 10 trames/s, soit  $\simeq 1.6$  kbits/tr ou encore  $\simeq 0.06$  bit/pixel au format QCIF), nous réalisons rapidement que ces résultats ne sont pas du tout satisfaisants. Nous devons orienter notre recherche dans une autre direction: elle portera sur l’étude du codage arithmétique binaire (BAC).

### 3.2.5. L’algorithme JBIG de codage arithmétique binaire

En 1988, un groupe d’experts vidéo est formé en vue de l’établissement d’un standard international pour le codage d’images binaires, appelé JBIG (Joint Bi-level Image Group). JBIG est sponsorisé par l’organisme ISO ((IEC/JTC1/SC2/WG9) et la CCITT. JBIG offre un codage sans perte. Comparée aux algorithmes G.3/G.4 mis

en œuvre pour le facsimile [22], la compression est améliorée, et si on le souhaite, la transmission peut être progressive. De plus, l'algorithme JBIG peut coder des images à plusieurs niveaux de gris, ainsi que des images couleur. Remarquons que, si un codage sans perte n'est pas exigé, le standard JPEG (Joint Picture Expert Group) fournit la meilleure compression.

### 3.2.5.1. La structure de JBIG

Cette section décrit la procédure de codage arithmétique binaire utilisée par JPEG et JBIG [26]. Ce codeur arithmétique JPEG/JBIG est aussi connu sous le nom de QM-codeur, descendant du Q-codeur [27], mais amélioré de manière significative au niveau des modules de codage arithmétique et d'estimation de probabilité. Globalement, le codeur JBIG peut être décomposé en une chaîne de  $D$  codeurs identiques, suivi d'un codeur en bout d'échelle. En Fig. 3.7,  $I_D$  dénote l'image à l'étage  $D$  et  $C_D$  sa version codée. Ce codage est progressif ou à multi-résolution: une image est représentée par une version basse-résolution plus une séquence de fichiers "delta" permettant de doubler la résolution. Quand une image, codée de façon progressive, est décodée, la version basse-résolution est en premier disponible. La valeur  $D$ , de doublements de résolution, libre dans le codeur JBIG, est généralement choisie entre 4 et 6. Elle peut aussi valoir 0, ce qui annule la notion progressive et non les avantages de compression.

Le cœur des codeurs de chaque niveau est de type arithmétique adaptatif. Les codeurs arithmétiques se distinguent des autres codeurs d'entropie, dans la mesure où ils transforment une chaîne de symboles à coder en un nombre réel appartenant à l'intervalle  $[0, 1[$ . Au lieu d'une suite de symboles, une représentation binaire de ce nombre va être transmis. Ce genre de processus est appelé division d'intervalles [28]. Une remarque importante pour ce type de codage est la suivante: il utilise



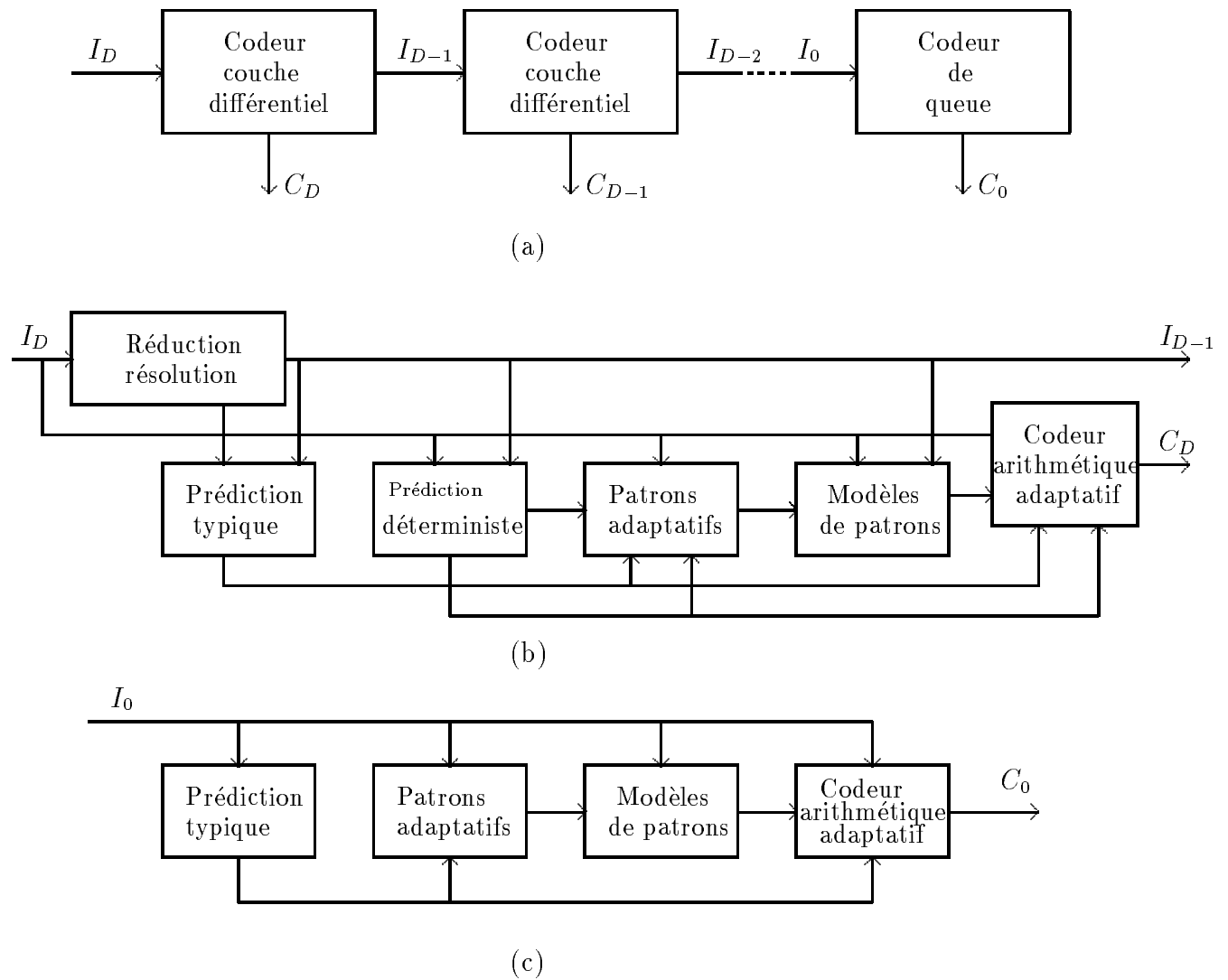


FIG. 3.7 - (a) Un codeur JBIG peut être décomposé en une chaîne de (b)  $D$  codeurs différentiels et d'un (c) codeur en queue

la propriété de “pipelining”, permettant au codeur de commencer à sortir les bits (à transmettre) d’une représentation binaire, avant d’avoir entièrement lu le flux à coder. Pour le décodeur, une propriété analogue reste vraie; il permet de sortir les symboles reconstruits avant d’avoir lu entièrement la représentation binaire à décoder.

Les différents principes du Q-codeur sont détaillés dans [27], et chaque aspect fondamental du codeur arithmétique binaire adaptatif est expliqué: codage arithmétique, procédures hardware et software, estimation de probabilité, etc. Nous allons plutôt porter notre attention sur les articles [28][29][30] proposant une version plus compacte d’un codage arithmétique. Dans la suite, nous allons décrire les outils que nous avons effectivement utilisés. La structure JBIG, jugée trop complexe pour être implantée en un temps raisonnable, a dû être simplifiée, et surtout, la partie d’estimation de probabilité a été remaniée. Dans les sections 3.2.5.2, 3.2.5.3 and 3.2.5.4, les modifications réalisées sont explicitées.

### 3.2.5.2. Codage arithmétique binaire (BAC)

L’algorithme BAC peut être utilisé pour coder n’importe quel ensemble d’événements, quelle que soit la forme de l’événement. Il transforme les événements en les codant en binaire sous forme de chaîne. Pour les images binaires, nous associons le pixel noir à l’événement le plus probable (T), et le pixel blanc au moins probable (F). Le codeur BAC a besoin d’estimer le taux relatif de (F) à la puissance de 2 la plus proche. Nous avons donc besoin de  $N_{sk}$  distincts paramètres  $sk$  de codage, appelés “skew numbers” (que l’on notera par la suite  $\mathcal{S}$ ), indexés de 1 à  $N_{sk}$ , nous indiquant la fréquence relative d’apparition de (F) (sorte de probabilité). En approximant à  $N_{sk}$  le nombre de  $\mathcal{S}$  au lieu d’utiliser des valeurs non discrètes, la perte maximale d’efficacité du codage dépasse à peine 4%.

L'algorithme BAC est initialisé avec l'intervalle-espace unité  $[0, 1[$ , exprimé par  $C = 0.0$  (point-code) and  $A = 1.0$  (espace-code disponible). Le codeur BAC va successivement diviser la taille de l'espace-code disponible  $A$  en deux sous-intervalles, selon la valeur courante  $sk$  de  $\mathcal{S}$ . L'intervalle de gauche est associé à (F), et celui de droite à (T). Les variables  $C$  et  $A$  décrivent conjointement l'intervalle qui représente en binaire les symboles codés. Comme pour l'espace-code initial, l'intervalle courant est de la forme  $[C, C + A[$  (cf. Fig. 3.8). De cette manière, le code arithmétique construit la chaîne codée par ajout d'un élément "summand" à la chaîne courante. Cette opération de sommation induit le problème de retenue ou "carry-over". Cependant, il est montré dans [29] comment une utilisation particulière de  $A$  et  $C$  et de fonctions tampons permet l'élimination des retenues. En réalité,  $A$  et  $C$  sont de précision finie, pas plus de 16 bits par exemple. Soient  $W(F)$  and  $W(T)$  étant respectivement les largeurs des sous-intervalles assignés à (F) et (T). La procédure comprend trois étapes:

1. Etant donné  $sk \in \mathcal{S}$ , on calcule

$$W(F) = 2^{-sk}$$

$$W(T) = A - 2^{-sk}$$

2. Etant donnés (T) ou (F),  $W(T)$  et  $W(F)$ ,

$$\text{si (T) : } C(s, T) = C(s) + W(F) \text{ et } A(s, T) = W(T)$$

$$\text{si (F) : } C(s, F) = C(s) \text{ et } A(s, F) = W(F)$$

3. Nous éliminons le problème des retenues en utilisant un tampon  $Q$  de type FIFO et de longueur arbitraire, qui va stocker au fur et à mesure la chaîne codée.

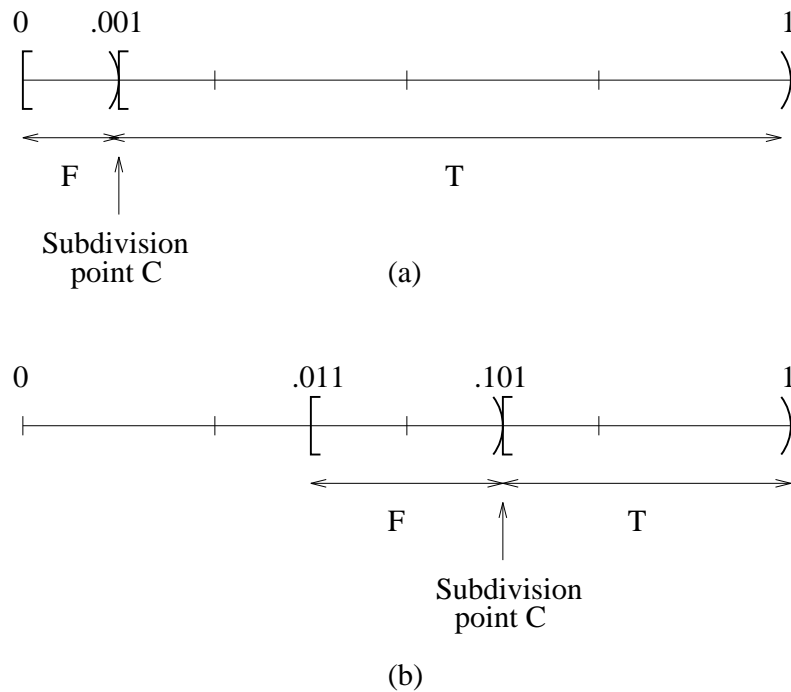


FIG. 3.8 - Fission de l'intervalle pour  $N_{sk} = 4$ , (a) avec (T) et  $sk = 3$ , (b) avec (T) et  $sk = 1$

L'algorithme du codeur BAC, avec la renormalisation intervenant lors des retenues, est décrit par l'organigramme de la Fig. 3.9. Les unités de statistique provenant des  $\mathcal{S}$ , sont limitées à  $1, 2, \dots, N_{sk} - 1$ . Le codeur accepte de l'information binaire d'événements successifs par la variable *MIN* (More probable IN). Les registres  $C$  et  $A$  ont une longueur de  $N_{sk}$  bits. Ils sont respectivement initialisés à  $0.\underbrace{00 \dots 0}_{N_{sk}-1}$  et  $1.\underbrace{00 \dots 0}_{N_{sk}-1}$ . Soit  $shl^p$  l'opération de translation de  $p$  bits vers la gauche. Au bout de la procédure, la chaîne codée est envoyée, pour le stockage, au tampon  $Q$  de type FIFO. Après que le dernier événement a été codé, il subsiste le dernier intervalle  $[C, C + A[$ . Si la longueur de l'information à coder est connue, nous savons quand le dernier événement est codé. En conséquence, toute chaîne-code dont l'amplitude appartient à  $[C, C + A[$  permet de décoder la chaîne de données originales. Il nous est alors possible d'effectuer la troncature des 0 en fin de chaîne codée, ce qui favorise le phénomène de compression.

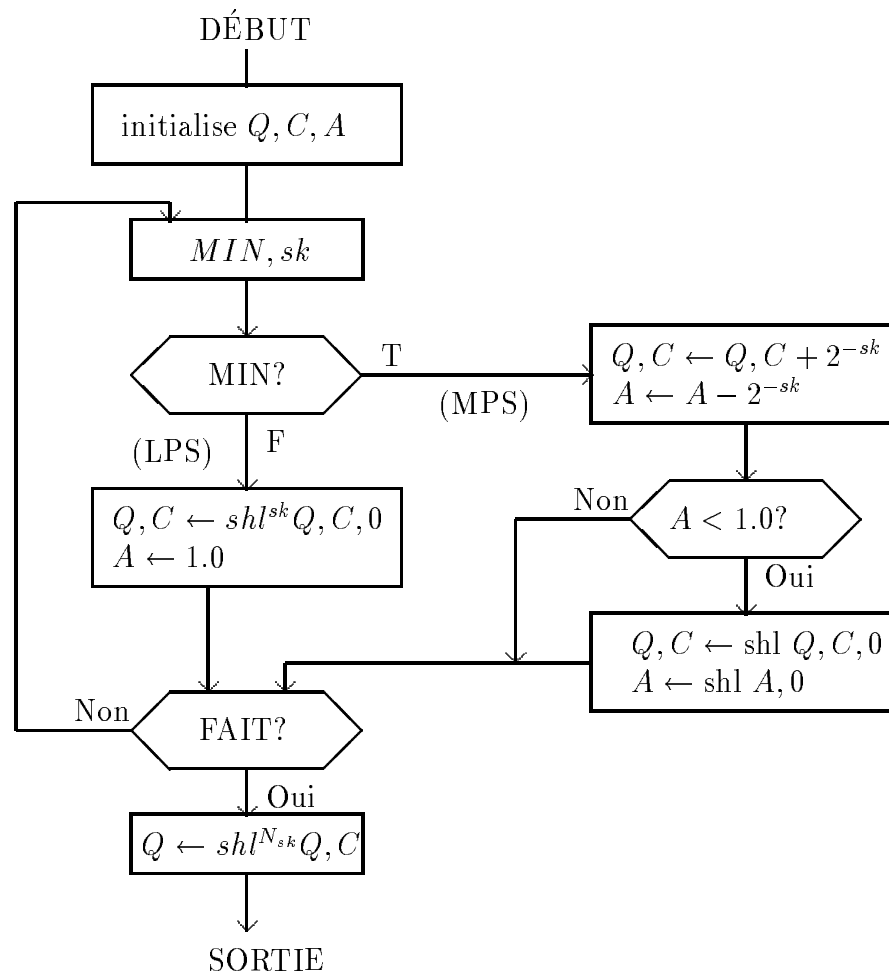


FIG. 3.9 - Organigramme du codeur arithmétique binaire

La partie de décodage par l'algorithme BAC est détaillée par l'organigramme de la Fig. 3.10. Le registre  $A$  est initialisé à  $1.\underbrace{00\dots0}_{N_{sk}-1}$  et  $C$  aux  $N_{sk}$  premiers bits en sortie du tampon  $Q$  de type FIFO. Quant à l'estimation des valeurs statistiques, il nous faut une bonne estimation des  $\mathcal{S}$ . Cette tâche est primordiale dans l'implantation de l'algorithme BAC. Avant de proposer notre propre système d'estimation, présentons celui que l'on rencontre dans le standard JBIG.

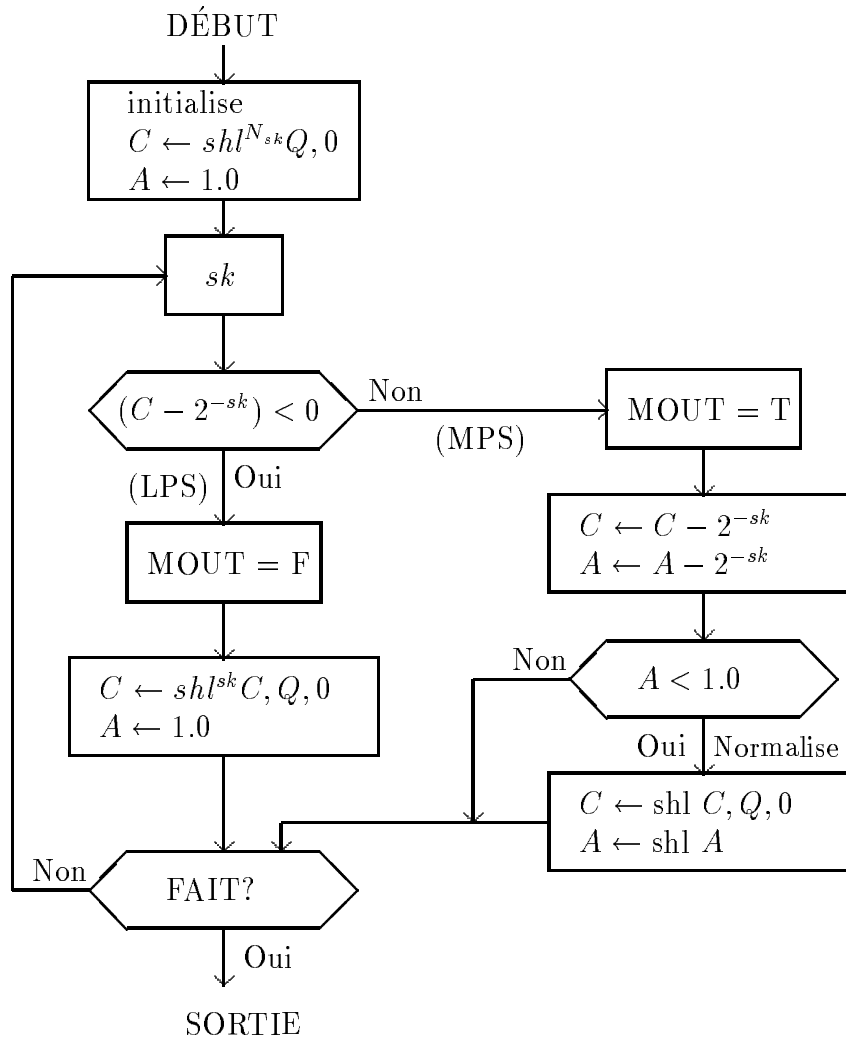


FIG. 3.10 - Organigramme du décodeur arithmétique binaire

### 3.2.5.3. Estimation de la probabilité

L'estimation adaptative de la probabilité a été utilisée dans la plupart des codeurs arithmétiques. Elle prend en compte les propriétés de la source à coder. Le processus d'estimation de la probabilité, implanté dans le QM-codeur, est basé sur une estimation à renormalisation développée dans le Q-codeur. Plus précisément, il s'appuie sur une forme de comptage approximatif dans laquelle la renormalisation des registres intervalle est prise en compte pour estimer les événements MPS (T) and LPS (F) comptés. Qu'il y ait ou non renormalisation pour ces deux symboles, le calcul d'apparition de MPS est toujours effectué, et une nouvelle estimation de sa probabilité est obtenue d'une table, fournissant une plus grande valeur  $Q_e = 2^{-sk}$  quand la renormalisation LPS apparaît, et une plus petite valeur  $Q_e$  dans le cas de l'apparition de MPS. Les décisions de  $Q_e$  sont régies par une machine à plusieurs états. En effet, la machine à état du processus d'estimation correspond à une chaîne de Markov, contenant un état pour chaque estimation de probabilité (Fig. 3.11). Excepté les deux états en bout de chaîne, les états peuvent transiter entre eux. Pour approfondir ces points, il faut se référer à [31] où chaque notion est présentée.

Cette description nous renseigne sur certains problèmes auxquels nous serions

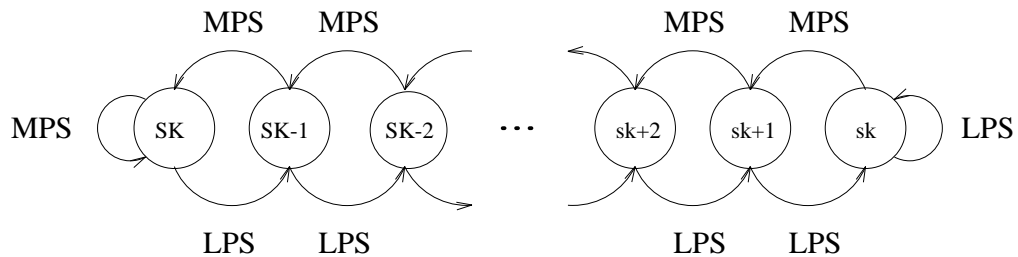


FIG. 3.11 - Diagramme d'état de l'estimateur de probabilité JBIG

confrontés si nous utilisons ce procédé classique d'estimation développé dans JBIG. En effet, il s'agit pour nous de coder une image binaire. Dans notre cas, l'événement LPS représentatif des états de transition est très peu probable. De ce fait, l'évolution

de  $Q_e$  est peu applicable dans ce cas, et la compression peut s'avérer catastrophique. C'est pourquoi, nous allons modifier le processus d'estimation en le simplifiant, et en utilisant un modèle de patron pour la décision des  $\mathcal{S}$ . D'après la Fig. 3.12, l'estimation de probabilité est réduite à une chaîne de Markov à seulement deux états. Ces deux états sont  $sk \in \{1, s = N_{sk} - 1\}$ . Ceci nous permet d'implanter assez simplement les organigrammes 3.9 and 3.10 de codage et décodage BAC, et de proposer une bonne procédure de compression d'images binaires.

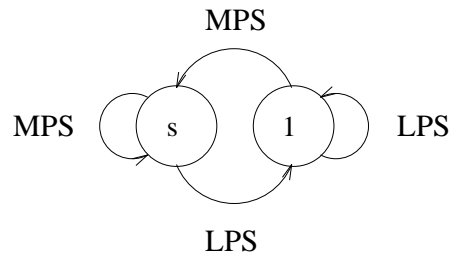


FIG. 3.12 - Diagramme d'état de notre unité statistique

#### 3.2.5.4. Modèle de patron utilisé

Les pixels déterminant le patron sont sélectionnés dans ceux déjà codés dans le voisinage immédiat du pixel courant. Comme le modèle de patron doit être causal, les échantillons doivent être à la fois connus au codeur et au décodeur. En considérant un modèle séquentiel de type JBIG, les échantillons disponibles sont situés sur la même ligne à gauche que le pixel courant, et sur les lignes précédentes (selon le sens du balayage). Comme aucun modèle n'a encore été proposé dans le cas qui nous occupe, nous définissons le patron de la Fig. 3.13. Afin d'appliquer ce patron aux images binaires disponibles, nous devons effectuer quelques pré-traitements: nous avons modifié les codes génériques 1, 2, 3 respectivement en mots-codes 1, 11, 111 (codage de Huffman modifié), nous permettant d'associer un bit par pixel, et un seul, puis d'appliquer le modèle de patron préalablement défini. Mais cette attribution d'un bit par pixel n'est envisageable que si nous supposons que deux transitions



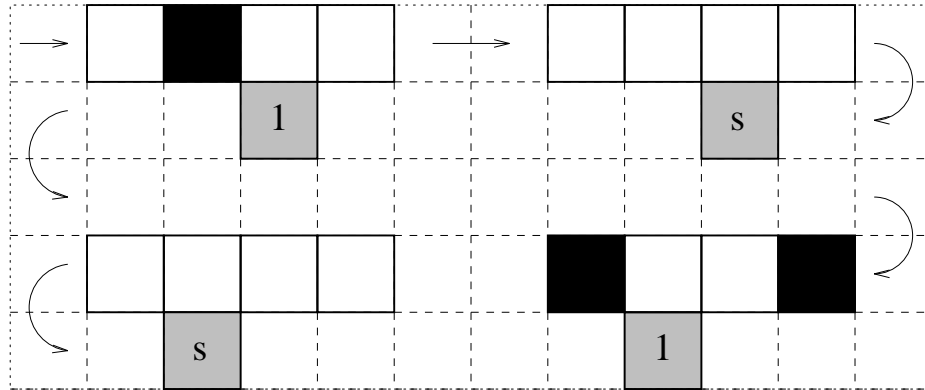


FIG. 3.13 - Modèle de patron pour la décision des  $\mathcal{S}$  avec le balayage proposé. En gris est représenté le pixel pour qui le  $\mathcal{S}$  est à déterminer.

horizontales consécutives ainsi que deux transitions séparées par un pixel ne sont pas permises. Cette hypothèse, qui doit être prise en compte dans la segmentation, est équivalente à interdire aux régions des longueurs de transition horizontale de 1 ou 2 pixels. Avec cette hypothèse, l'unique décodabilité des trois mots-codes 1, 11, 111 est alors assurée.

Ces hypothèses et modifications sont faites dans le but de simplifier l'implantation de l'algorithme JBIG en C++, tout en espérant que les résultats obtenus seront proches de ceux produits par le vrai système JBIG. Pour l'exemple de la Fig. 3.14, le début du flux à coder par BAC devient:

```
11000000000000000000000000000011010000011
00010000000010000000010111001000000010...
```

En dehors des problèmes liés à l'utilisation de l'algorithme BAC pour ce projet, ce type de codage devrait donner de meilleurs résultats qu'un simple codage de Huffman ou de RLC. En outre, l'information codée est représentée de manière compacte et la création de modèles de patrons et d'unités statistiques rendent l'implantation efficace [32].

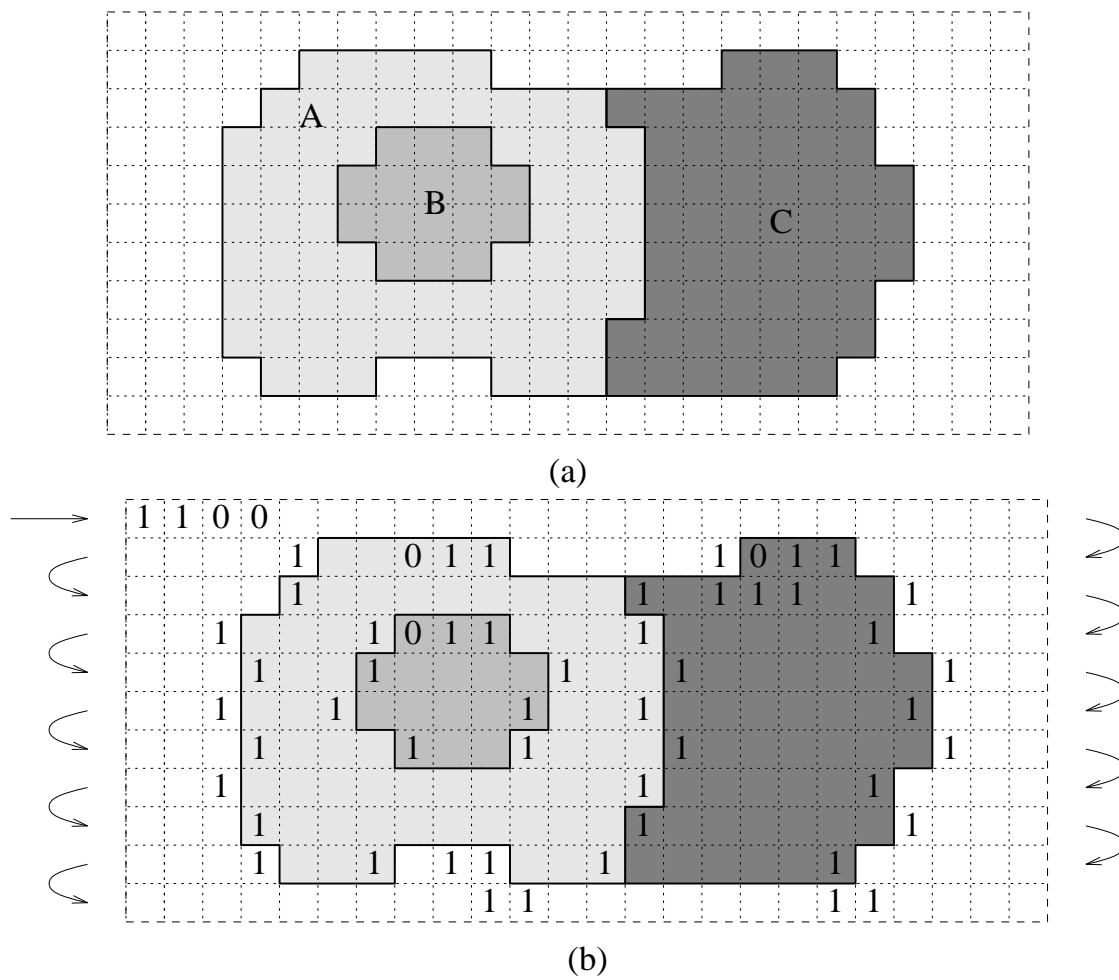


FIG. 3.14 - Exemple d'une (a) segmentation avec (b) les différents états de transition associés au codage BAC

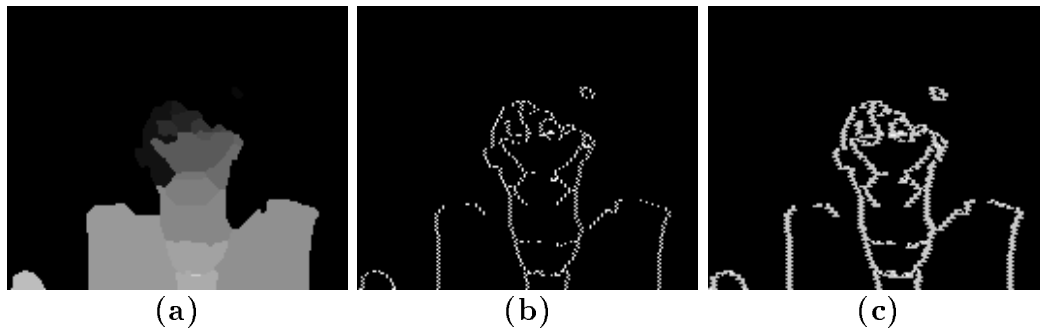


FIG. 3.15 - Exemple de BAC: (a) une trame segmentée, (b) extraction des états de transition et (c) représentation binaire des  $\mathcal{S}$

### 3.2.6. Conclusion

	bits	bit/pixel
“Miss America”	2168	0.086
“Carphone”	6871	0.271
“Foreman”	8356	0.330

TAB. 3.6 - Nombre de bits, pour la trame #0 segmentée de chaque séquence, par codage arithmétique binaire simplifié (BAC)

Les deux techniques proposées sont intéressantes car elles permettent de coder une image en respectant un simple balayage. Mais sont-elles bien adaptées pour coder des telles images binaires? Rien n’est sûr. Les résultats obtenus ne sont pas aussi probants que ceux auxquels on pouvait s’attendre; dans le cas du BAC, la compression est améliorée, mais reste encore bien en dessous des buts fixés (Tableau 3.6). De plus, l’étape d’estimation des unités statistiques du BAC ainsi que le processus de reconstruction restent complexes. Bien qu’il soit possible d’apporter quelques améliorations à ces deux codages, nous restons sceptiques sur leurs aboutissements. C’est pourquoi, nous allons nous orienter vers un autre modèle de codage qui peut apporter des solutions satisfaisantes au problème de compression des cartes de segmentation. Il s’agit de la méthode de codage par chaînes.

## 3.3. Le codage par chaînes (CC)

Dans ce chapitre, nous nous intéressons aux méthodes se proposant d’extraire le contour entier des cartes de segmentation, et non pas seulement certains états de transition. Cette approche a été exposée et mise en œuvre dans un grand nombre de publications, telles que [13][33][34], et semble parfaitement adaptée aux problèmes de codage vidéo à très bas débit. La technique la plus connue de codage des contours est le codage par chaînes. De nombreux articles décrivent cette technique et ses successives améliorations. Les papiers [35][36][37][38][39] sont parmi les références

majeures relatives à cette méthode.

### 3.3.1. Description de l'algorithme CC

Les principales étapes du CC, pour le codage des cartes de segmentation, sont les suivantes:

1. *Extraction du contour intérieur des régions.*

La première étape consiste à extraire le contour intérieur de chaque région. Nous utilisons un modèle de connectivité par quatre pour définir la topologie liée au modèle du contour (Fig. 3.16(a): il suffit que l'un des quatre pixels voisins soit différent pour que le pixel courant devienne un point de contour). Afin de mieux comprendre ce genre de codage, un exemple est proposé à la Fig. 3.18. Nous considérons trois régions: les régions *A* et *C* sont accolées l'une à l'autre, et *B* est entièrement contenue dans *A*. Chaque contour, extrait grâce au modèle de voisinage, est fermé. En outre, dès que deux régions sont en contact, le contour mitoyen est extrait deux fois. Mais il est évident qu'il devra être codé une seule fois, sinon nous risquons de perdre l'efficacité du CC. Comme les régions sont amenées à se toucher ou s'inclure, nous devons déterminer un ordre de codage des chaînes, dépendant de la position de chaque région vis-à-vis de ses voisines.

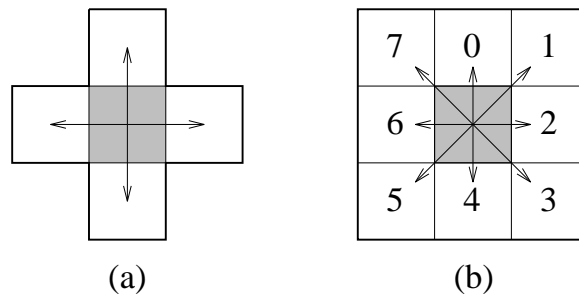


FIG. 3.16 - Modèles de voisinage (a) par quatre pour l'extraction des contours, (b) par huit pour le suivi des contours extraits

## 2. *Décision d'un ordre de codage des chaînes.*

Cette opération est primordiale pour deux raisons majeures: les régions contenant d'autres régions doivent être impérativement codées avant ces dernières. De plus, il faut absolument éviter de coder deux fois des parties de chaîne correspondant à des frontières entre régions. Pour cela, nous dénombrons, pour chaque région, le nombre de ses régions voisines. La frontière de l'image est considérée comme un contour particulier. En triant par ordre décroissant l'ensemble de ces nombres de voisins, nous déterminons l'ordre de codage des chaînes. Cette façon de procéder est rapide et efficace, puisque le nombre de régions voisines de chaque région, est un entier petit. Maintenant que l'ordre de codage a été décidé, il nous faut définir les points de départ et d'arrêt de chaque chaîne à coder.

## 3. *Détermination des points de départ et d'arrêt d'une chaîne.*

Cette étape sert, au maximum, à éviter de coder deux fois des mêmes parties de chaîne. Pour cela, nous devons définir quelles sont les parties de chaînes à retenir pour le codage. La Fig. 3.18 représente un cas d'espèce concernant la détermination des points de départ des trois chaînes, l'ordre de codage et le codage effectif des bouts de chaîne. Pour une région isolée (sans voisin ou strictement incluse dans une autre région), nous choisissons comme point de départ de la chaîne le pixel le plus haut à gauche qui appartient au contour extrait. Par contre, si la région possède plus d'un voisin, ou bien s'étend jusqu'à la frontière de l'image, le point de départ et le point d'arrêt sont définis de telle sorte que le bout de chaîne, défini entre ces deux pixels, corresponde à une partie de chaîne déjà codée. De plus, parmi toutes les possibilités de tels couples de points, nous retiendrons celle où leur distance mutuelle est maximale.

#### 4. Codage des directions.

En Fig. 3.17, nous avons assigné à chaque pixel (du point de départ à celui précédant le point d'arrêt) la direction (Fig. 3.16(b)) pour obtenir le point suivant de la chaîne. Nous obtenons ainsi pour chaque chaîne une suite de directions représentant les contours de cartes de segmentation, c'est à dire l'information de segmentation.

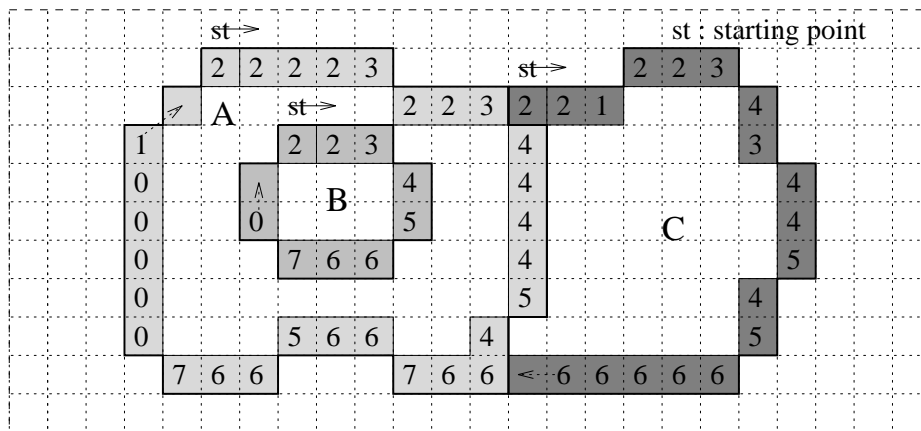


FIG. 3.17 - Exemple d'attribution des directions dans le "chain coding"

Toujours pour le même exemple, les suites de directions des régions A,B,C sont:

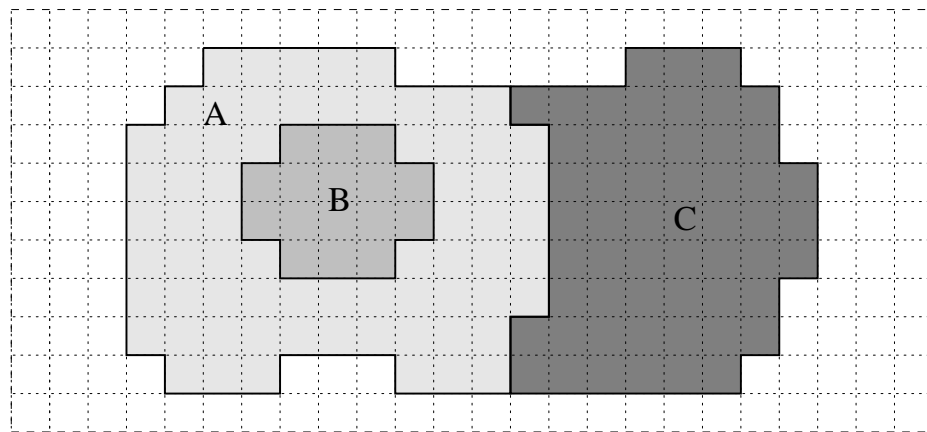
A: 22223223444454667665667000001

B: 223456670

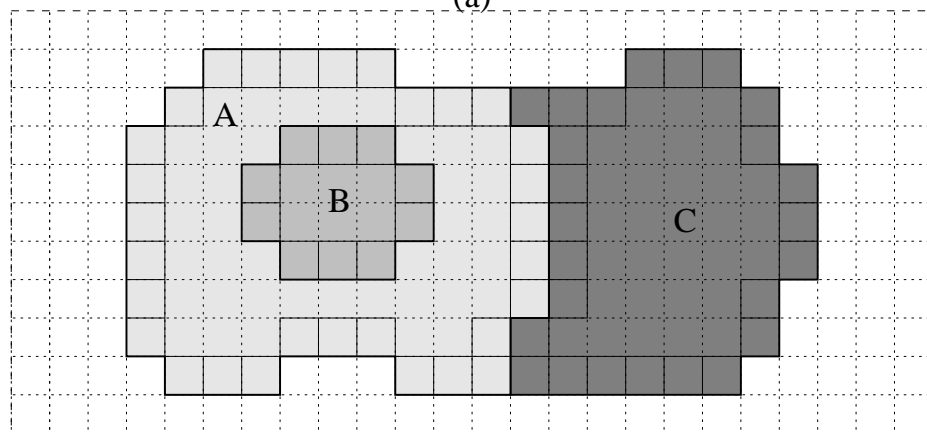
C: 221223434454566666

### 3.3.2. Codage effectif des chaînes

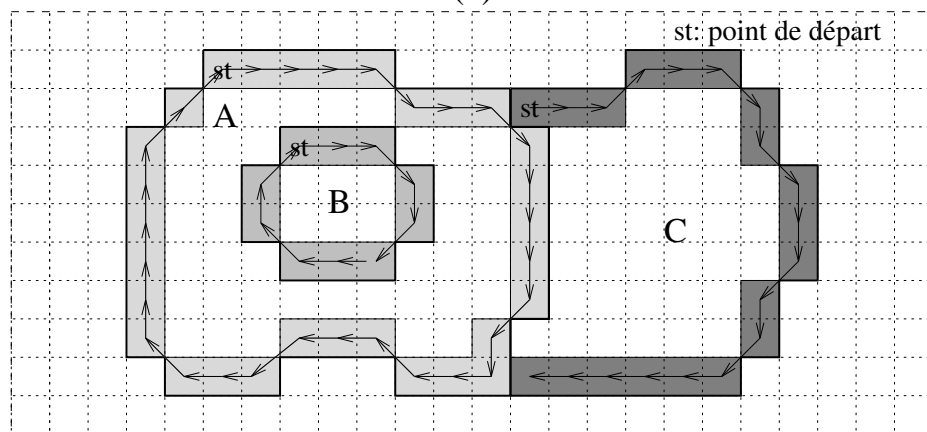
D'après l'exemple des suites de directions décrit plus haut, il existe une corrélation significative entre deux directions successives. Afin d'effectuer une décorrélation nécessaire pour un codage efficace, nous nous sommes intéressés à deux formes de codage des suites de directions des chaînes: l'un est de type prédictif et l'autre de



(a)



(b)



(c)

FIG. 3.18 - Exemple de codage de contour par la méthode CC: (a) carte de segmentation, (b) extraction des contours et (c) codage des bouts de chaînes avec l'ordre A,C,B

type conditionnel.

### Codage prédictif

Le codage prédictif consiste simplement à calculer la différence entre deux directions consécutives. Il peut être formulé de la manière suivante: si nous notons par  $X_0, X_1, \dots, X_{N-1}$ , les symboles d'une source  $X$  (chaîne), identiquement distribués, alors nous sommes amenés à poser

$$D_i = \begin{cases} X_i - X_{i-1}, & i = 1, \dots, N - 1 \\ X_0, & i = 0 \end{cases}$$

Les directions précédentes deviennent:

A: 20001701100017201707101100001

B: 201111011

C: 207101171017110000

Maintenant, nous supposons que les  $\{D_i\}$  sont indépendants et identiquement distribués (i.i.d). En assignant un code de Huffman à chaque code différentiel, toujours basé sur une table de Huffman construite à l'aide de l'histogramme des  $\{D_i\}$ , nous codons l'information différentielle des directions de chaîne.

### Codage conditionnel

Observons à nouveau la corrélation des suites de directions: dans le cas de contours assez longs, il existe des séries de symboles identiques intervenant régulièrement. Cette propriété est assurée tant que le contour est un minimum "lisse". Aussi, faisons l'hypothèse que  $X$  est un processus de Markov d'ordre 1 stationnaire, i.e. que les densités de probabilité conditionnelle satisfont

$$P(X_i = x_i | X_{i-1} = x_{i-1}, \dots, X_0 = x_0) = P(X_i = x_i | X_{i-1} = x_{i-1}), i = 1, \dots, N - 1$$

Ceci signifie que le processus de Markov est entièrement défini par une densité initiale  $P_{X_0}$  et les densités de transition  $P(X_i = x_i | X_{i-1} = x_{i-1})$ . De plus, il est intéressant de



formuler l'entropie conditionnelle de deux sources  $H_{co}(Y|X)$ , en incluant l'hypothèse nécessaire d'ergodicité:

$$\begin{aligned}
H_{co}(X_n|X_{n-1}) &= \sum_{x_{n-1} \in X_{n-1}} H(X_n|X_{n-1} = x_{n-1})P(X_{n-1} = x_{n-1}) \\
&= - \sum_{x_{n-1} \in X_{n-1}} \sum_{x_n \in X_n} P(X_n = x_n|X_{n-1} = x_{n-1})P(X_{n-1} = x_{n-1}) \cdot \\
&\quad \log_2 P(X_n = x_n|X_{n-1} = x_{n-1}) \\
&= - \sum_{x_{n-1} \in X_{n-1}} \sum_{x_n \in X_n} P(X_n = x_n, X_{n-1} = x_{n-1}) \log_2 P(X_n = x_n|X_{n-1} = x_{n-1})
\end{aligned}$$

De plus, si nous définissons le taux d'entropie  $H_\tau$  d'un processus stochastique  $\{X_i\}$  par

$$H_\tau = \lim_{n \rightarrow \infty} \frac{1}{n} H(X_0, X_1, \dots, X_{n-1}),$$

il est montré dans [40] que, pour un processus de Markov d'ordre 1 stationnaire, ce taux d'entropie vaut:

$$\begin{aligned}
H_\tau &= \lim_{n \rightarrow \infty} \frac{1}{n} H(X_0, X_1, \dots, X_{n-1}) \\
&= \lim_{n \rightarrow \infty} H(X_{n-1}|X_{n-2}, \dots, X_0) \text{ (stationnarité)} \\
&= \lim_{n \rightarrow \infty} H(X_{n-1}|X_{n-2}) \text{ (processus de Markov-1)} \\
&= H_{co}
\end{aligned}$$

Comparons le résultat de la quantité  $H_{co}$  avec la longueur moyenne  $L_{co}$  du code, exprimée par

$$L_{co} = \sum_{x_{n-1} \in X_{n-1}} \sum_{x_n \in X_n} P(X_n = x_n, X_{n-1} = x_{n-1}) l(X_n = x_n|X_{n-1} = x_{n-1}),$$

les valeurs  $l(\cdot|\cdot)$ , étant égales aux longueurs des mots-codes respectivement associés aux probabilités conditionnelles  $P(\cdot|\cdot)$ , sont fournies par la table de Huffman construite à partir des  $P(\cdot|\cdot)$ . Les 64 états conditionnels possibles ont aussi été déterminés

sur les trois séquences de référence “Miss America”, “Foreman”, “Carphone”. Les histogrammes des probabilités conditionnelles de chaque séquence sont représentés aux Tableaux 3.7, 3.8 et 3.9 ( $P(X_n|X_{n-1})$  avec  $X_{n-1}$  en première colonne et  $X_n$  en première ligne). Sur ces trois Tableaux, on utilise alors un code de Huffman différent pour chacune des rangées (lignes).

	0	1	2	3	4	5	6	7
0	72	13	2					13
1	22	50	28					
2		12	71	14	3			
3			25	40	30	5		
4				15	66	15	4	
5					38	41	21	
6	1					11	74	14
7	42	1					15	42

TAB. 3.7 - Histogramme (%) des états conditionnels  $X_n|X_{n-1}$  pour “Miss America” ( $\hat{H}_{co} = 1.07$ )

	0	1	2	3	4	5	6	7
0	71	13	2					14
1	29	37	34					
2		14	74	7	5			
3			17	42	41			
4				16	71	10	3	
5					31	27	42	
6	4					7	84	5
7	40						30	30

TAB. 3.8 - Histogramme (%) des états conditionnels  $X_n|X_{n-1}$  pour “Carphone” ( $\hat{H}_{co} = 1.16$ )

### 3.3.3. Codage du point de départ de chaque chaîne

Nous codons le point de départ de chaque chaîne avec 16 bits. Cependant, cette quantité de bits alloués au premier pixel est excessive, et de plus cette allocation se

	0	1	2	3	4	5	6	7
0	74	7	5					14
1	9	28	63					
2		30	42	25	3			
3			33	57	10			
4				14	68	7	11	
5					18	33	49	
6	4					25	51	20
7	11						31	58

TAB. 3.9 - Histogramme (%) des états conditionnels  $X_n|X_{n-1}$  pour “Foreman” ( $\hat{H}_{co} = 1.17$ )

répète pour chaque chaîne, i.e., chaque région. Par exemple, lors d’une segmentation intra-trame, chaque trame de la séquence “Miss America” nécessite en moyenne une vingtaine de chaînes à coder. Par ailleurs, les séquences “Carphone” ou “Foreman” en nécessitent généralement plus d’une cinquantaine. De façon grossière, la quantité de bits allouée à l’ensemble des points de départ des chaînes correspond à 20% du nombre total de bits pour le codage des chaînes. Ce pourcentage est effectivement excessif.

### 3.3.4. Reconstruction par remplissage intérieur

Le remplissage de l’intérieur d’une région, dont le contour est connu, est l’un des problèmes classiques de reconstruction dans les applications de traitement d’images, de reconnaissance de formes et de vision par ordinateur. On le résoud à l’aide de différents algorithmes qui calculent des intégrales sur des parties de régions, et donc requièrent la connaissance de l’intérieur des régions. Les principaux algorithmes existants, pour le remplissage de régions, sont les techniques de contrôle de parité [41], les techniques de remplissage basé sur la connectivité [42], et les méthodes se servant du codage par chaînes des contours (par exemple [43]). Les deux premières solutions citées possèdent beaucoup trop d’inconvénients dans le traitement de cas

particuliers. Ici, nous allons introduire un algorithme très simple et très efficace, de remplissage de l'intérieur de régions fermées [44]. Nous avons vu plus haut que nous

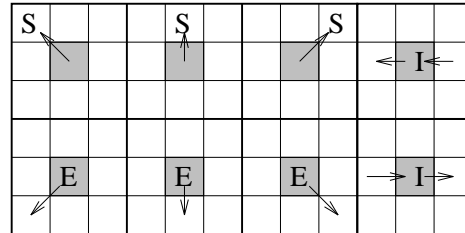


FIG. 3.19 - Attribution des codes S, E et I dans le processus de reconstruction

assignons, à tout pixel d'une chaîne à coder, une des huit directions (ici de 0 à 7: voir Fig. 3.16(b)). Si nous décidons de faire le remplissage, par le balayage habituel allant du haut vers le bas et de la gauche vers la droite de l'image, nous remarquons que les codes-directions, 7, 0 et 1, deviennent les points de départ de gauche pour le remplissage de la région. Il en est de même pour les codes-directions 3, 4 et 5 qui représentent les point d'arrêt du remplissage. En tenant compte de cette remarque, un codage binaire des contours peut être proposé: il suffit d'assigner le code "S" au point de départ de remplissage ayant l'une des trois premières directions citées, et d'assigner le code "E" au point d'arrêt du remplissage. Dans ce processus de codage, "S" doit être codé sur le pixel courant, tandis que "E" doit être codé sur le pixel précédant le pixel courant (voir Fig. 3.19). Si un pixel, venant juste d'être codé avec "S", doit être codé avec "E", ou inversement, nous changeons le code par le code "I". Après assignation des trois codes sur les pixels de contour, le processus de reconstruction devient évident; en commençant du pixel de la région situé le plus en haut à gauche, dès que nous rencontrons un pixel codé par "S", nous remplissons les pixels de la ligne jusqu'à tant d'atteindre un pixel codé par "E". Les pixels codés par "I" sont purement et simplement ignorés (Fig. 3.20(b) et (c)).

Dans le cas d'une chaîne ouverte, nous devons d'abord la fermer, en suivant

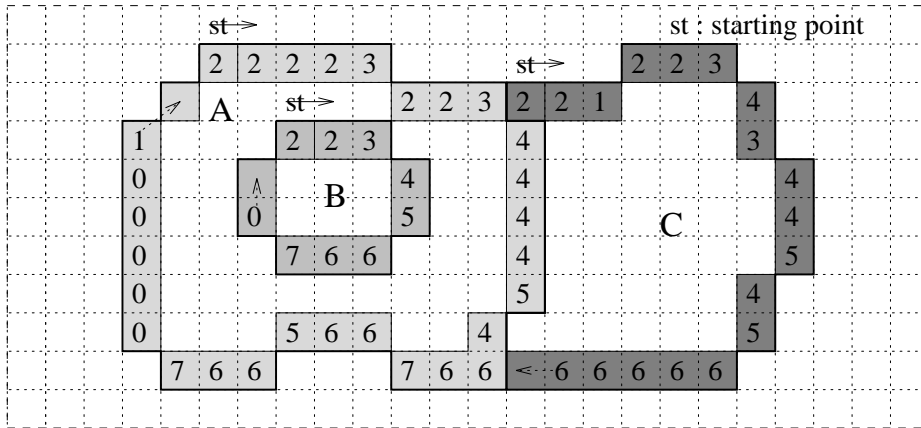
les extrémités de chaîne déjà décodées appartenant aux frontières entre régions. En effet, le processus de reconstruction s'effectue dans le même ordre de codage des chaînes. De ce fait, toute extrémité de chaîne manquant à la région courante a déjà été décodée et la continuité entre frontières n'est pas perdue. Dans le cas de régions incluses dans d'autres (ex: C dans A), la région incluse est reconstruite après celle qui la contient.

### 3.3.5. Conclusions

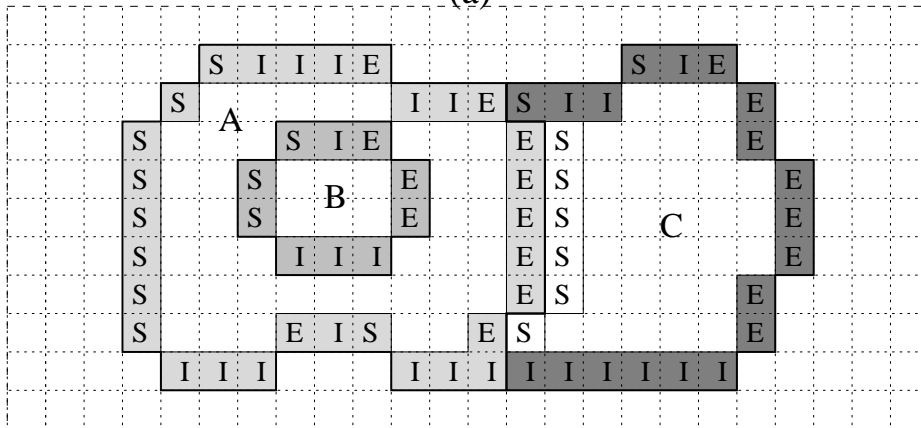
	bits	bit/pixel
"Miss America"	1539	0.061
"Carphone"	4604	0.182
"Foreman"	5606	0.221

TAB. 3.10 - Nombre de bits, pour la trame #0 segmentée de chaque séquence, par codage CC

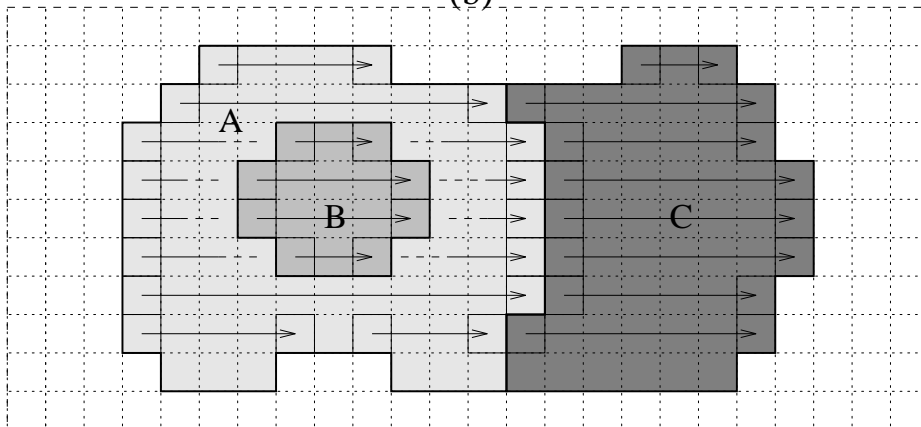
La méthode CC est très efficace car chaque étape décrite s'implante aisément. Par cette technique, les meilleurs résultats concernant le taux de compression ont été obtenus (Tableau 3.10). Dans le cas de la séquence "Miss America", les taux obtenus sont bien plus proches de ceux qui nous intéressent pour un codage à très bas débit ( $\simeq 0.06$  bit/pixel). Les résultats donnés au Tableau 3.10 sont adaptés à chaque séquence d'images. Si nous utilisons un code de Huffman pour transmettre les mots-codes au récepteur, le coût s'élève à 280 bits par séquence, ce qui reste non négligeable. Notons que, dans le cas de l'utilisation d'un code unique, nous obtenons un léger excédent de bits par trame par rapport aux résultats présentés. Le code unique au récepteur est donc recommandé. Bien entendu, des améliorations au codage CC peuvent encore être apportées. L'une d'entre elles concerne le codage du point de départ de chaque chaîne, pour lequel le problème posé est encore en attente d'une solution pleinement satisfaisante. Nous aurions aussi pu utiliser un processus de Markov d'ordre  $p$ , avec  $p > 1$ , pour le modèle conditionnel de codage,



(a)



(b)



(c)

FIG. 3.20 - Exemple de reconstruction: (a) décodage des directions, (b) attribution des codes S, E et I et (c) processus de reconstruction complété

mais un nombre excessif de données doit alors être stocké.





# Chapitre 4

## Codage avec perte

D'après le chapitre précédent, consacré à l'étude de quelques techniques de codage sans perte, les résultats obtenus impliquent certaines conclusions évidentes; il semble que le codage CC est le plus efficace car il permet de d'atteindre pour certaines séquences les taux de compression envisagés. De plus, en comparant une estimation de l'entropie à une estimation de la longueur du code dans le cas du codage CC, nous remarquons que la marge de manœuvre est plutôt restreinte (voir chapitre 5). C'est pourquoi, il nous semble intéressant de nous tourner vers le codage avec perte pour en évaluer les possibilités et les avantages éventuels.

### 4.1. Enoncé de l'algorithme

Nous suivons les mêmes étapes que pour le codage CC; nous proposons là aussi d'extraire les contours de l'image, mais au lieu de considérer le codage de la totalité des chaînes, nous allons approximer les contours par des fonctions linéaires, et coder le résultat de l'approximation. Dans le cadre d'un codage à très bas débit, nous avons besoin d'une bonne méthode d'approximation de contours. Or, les auteurs de l'article [45] recommandent une technique efficace qui effectue une approximation

polygonale optimale des courbes: étant donnés  $N$  points ordonnés dans un plan, et une constante  $M < N$ , ils proposent un algorithme sélectionnant  $M$  points parmi les  $N$  donnés, qui définissent une courbe polygonale qui soit l'approximation linéaire par parties, continue du premier ordre et globalement optimale des  $N$  points. De plus, l'algorithme utilise un critère d'erreur choisi par l'utilisateur, et l'approche de programmation dynamique pour optimiser le temps de calcul souhaité dans de nombreuses situations pratiques.

### 4.1.1. Formulation du problème et algorithme de programmation dynamique

#### 4.1.1.1. Formulation du problème

Nous allons utiliser les notations suivantes pour la caractérisation du problème. Soit une séquence  $s$  finie de points de  $\mathcal{R}^2$ :

$$\begin{aligned} s &= s_1 s_2 s_3 \cdots s_{|s|} \\ &= (x_1, y_1)(x_2, y_2)(x_3, y_3) \cdots (x_{|s|}, y_{|s|}) \end{aligned}$$

Une courbe polygonale  $v$  sur  $s$  est définie comme une séquence croissante d'indices tels que les points de  $s$  associés sont les sommets de  $v$ :

$$v = v_1 v_2 v_3 \cdots v_{|v|}$$

Soit  $V_s^m$  l'ensemble des courbes polygonales fermées de  $s$  possédant  $m$  sommets. Si  $v_k$  et  $v_{k+1}$  sont deux sommets consécutifs de  $v \in V_s^m$ , alors la notation  $e(v_k, v_{k+1})$  représente l'arête entre  $s_{v_k}$  et  $s_{v_{k+1}}$ . L'arête englobe donc tous les points de  $s$  intermédiaires:  $s_{v_k}, s_{v_k+1}, \cdots, s_{v_{k+1}-1}, s_{v_{k+1}}$ . Nous pouvons associer une erreur ou mesure de distorsion de chaque arête, comme son degré de finesse par rapport aux points intermédiaires. Si  $v_k = i$  et  $v_{k+1} = j$ , notons par  $\Delta(i, j)$  une telle erreur (voir

section 4.1.3):

$$\Delta(i, j) = \sum_{k=i}^j d(s_k, e(i, j)), 1 \leq i \leq j \leq |s|;$$

où  $d(s_k, e(i, j))$  est une distorsion élémentaire du point  $s_k$  par rapport à l'arête considérée.

Le problème est alors le suivant: trouver une courbe polygonale fermée  $v \in V_s^M$  qui minimise la somme des erreurs définies sur ses arêtes. Formellement, nous cherchons à minimiser l'expression

$$\begin{aligned} \hat{D}(s, M) &= \min_{v \in V_s^M} \sum_{n=1}^{M-1} \Delta(v_n, v_{n+1}) \\ &= \min_{v \in V_s^M} \sum_{n=1}^{M-1} \sum_{k=v_n}^{v_{n+1}} d(s_k, e(v_n, v_{n+1})) \end{aligned} \quad (4.1)$$

Une solution à ce problème est donnée par le théorème suivant [45]:

**Théorème 1.** *Soient  $s$  une séquence de points et  $N = |s| \geq 2$ . Soit  $\hat{D}(s, M)$  définie comme en (4.1) et  $D(n, m)$  une fonction récursive vérifiant:*

$$D(n, m) = \begin{cases} \min_{m-1 \leq i \leq n-1} (D(i, m-1) + \Delta(i, n)), & n \geq m > 1 \\ 0, & n = m = 1 \\ +\infty, & \text{autrement} \end{cases}$$

Alors,  $\forall M > 1$ ,  $\hat{D}(s, M) = D(N, M)$ .

Le résultats de ce théorème permet l'interprétation suivante: si l'on se donne un nombre  $M$  de vertex, nous obtenons toutes les approximations polygonales optimales utilisant  $m$  vertex, pour tout  $m < M$ . Ceci nous donne une grande souplesse dans le choix du nombre de vertex par chaîne approximée.

#### 4.1.1.2. Algorithme de Programmation Dynamique (PD)

Des solutions informatiques à la fonction récursive du Théorème 1 peuvent être obtenues par programmation dynamique. Citons celle retenue dans [45]:

```

D(1, 1) = 0;
For n = 2 to N do D(n, 1) = +∞; EndFor;
For m = 2 to M do
  For n = m to N do
    D(n, m) =  $\min_{m-1 \leq i \leq n-1} (D(i, m-1) + \Delta(i, n));$ 
  EndFor;
EndFor;
ErreurTotale = D(N, M);

```

En utilisant un stockage approprié, une implantation itérative de l'algorithme est envisageable et elle réalise les calculs sur  $N \cdot M$  points au plus. On met en place un processus incrémental pour calculer les mesures de distorsion, ce qui n'augmente pas la complexité d'exécution. Pour ce type d'incrémental, le coût de l'algorithme vaut  $\Theta(MN^2)$ .

#### 4.1.2. Extraction des sommets de la courbe polygonale

Bien que l'algorithme évalue l'erreur totale d'approximation liée à une mesure d'erreur donnée, il nous est possible d'extraire facilement les  $M$  vertex de la courbe approximée. En effet, nous pouvons écrire que tout index  $v_p$  de sommet, tel que  $p \in \{M-1, \dots, 1\}$  est donné par

$$v_p = \arg_{\{i\}} \left\{ \min_{p \leq i \leq v_{p+1}-1} (D(i, p) + \Delta(i, v_{p+1})) \right\}$$

avec  $\{s_{v_1}, s_{v_M}\} = \{s_1, s_N\}$ .

Cette procédure d'extraction des vertex n'augmente pas le temps asymptotique de calcul du programme.

### 4.1.3. Mesures de distorsion

L'erreur introduite en substituant la série de points  $M_i(x_i, y_i)$ ,  $M_{i+1}(x_{i+1}, y_{i+1})$ ,  $\dots$ ,  $M_j(x_j, y_j)$ , au segment  $[M_i, M_j] = e(i, j)$  peut être calculée de divers moyens. Nous avons implanté deux critères d'erreur, en respectant la volonté d'incrémentation sous-entendue dans les boucles de l'algorithme.

1. *Critère d'erreur maximale (ME).*

Ici, l'erreur est mesurée comme la distance maximale entre le segment, défini par deux sommets consécutifs de l'approximation, et les points intermédiaires de la chaîne (Fig. 4.1 (a)) :

$$\Delta(i, j) = \max_{i+1 \leq k \leq j-1} \frac{|u_{i,j}x_k + v_{i,j}y_k + h_{i,j}|}{\sqrt{u_{i,j}^2 + v_{i,j}^2}} \text{ où}$$

$$u_{i,j} = y_j - y_i,$$

$$v_{i,j} = x_i - x_j,$$

$$h_{i,j} = x_jy_i - y_jx_i$$

2. *Critère de l'erreur moyenne quadratique (MSE).*

Ici, l'erreur est mesurée comme la somme au carré des distances entre le segment, défini par deux sommets consécutifs de l'approximation, et les points intermédiaires de la chaîne (Fig. 4.1 (b)) :

$$\Delta(i, j) = \sum_{k=i+1}^{j-1} \frac{(u_{i,j}x_k + v_{i,j}y_k + h_{i,j})^2}{u_{i,j}^2 + v_{i,j}^2} \text{ où}$$

$$u_{i,j} = y_j - y_i,$$

$$v_{i,j} = x_i - x_j,$$

$$h_{i,j} = x_jy_i - y_jx_i$$

Lors de l'exécution des programmes, nous remarquons qu'il n'y a guère de différence entre ces deux critères. En conséquence, nous utiliserons le critère MSE dans les

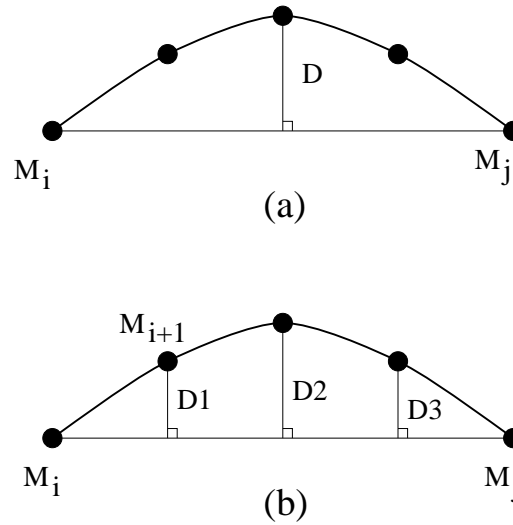


FIG. 4.1 - Mesures d'erreur: (a)  $\Delta(i, j) = D$  (ME) et (b)  $\Delta(i, j) = D1^2 + D2^2 + D3^2$  (MSE) entre deux sommets  $M_i$  et  $M_j$

simulations.

Nous définissons aussi le taux de distorsion de l'approximation pour une image par

$$\delta(\Delta) = \frac{1}{R} \cdot \sum_{r=0}^{R-1} \left( \frac{1}{M(r)} \sum_{n=0}^{M(r)-2} \Delta(v_n^r, v_{n+1}^r) \right) \quad (4.2)$$

si  $\{v_n^r\}_{n=0}^{M(r)-1}$  définit les  $M(r)$  sommets de la courbe polygonale d'approximation à la région  $r$ , avec  $r \in \{0 \cdots R-1\}$ . En fait, ce taux de distorsion dépend d'un facteur d'approximation  $\gamma$ . Ce facteur  $\gamma$  nous permet, en le divisant par la longueur de la chaîne à approximer, de déterminer une borne supérieure du nombre de sommets de la courbe polygonale.

## 4.2. Codage de la chaîne approximée

Cette méthode présente un intérêt particulier. D'après ce qui a été exposé, nous pouvons contrôler deux paramètres essentiels: l'erreur totale d'approximation

calculée à partir du second critère d'erreur précédemment défini, et le nombre de vertex de la chaîne approximée. En modifiant ces deux entités, le taux de compression peut être ajusté, et l'on contrôle ainsi le nombre de bits à réserver pour une chaîne. Les simulations vont confirmer ce point primordial dans ce procédé de codage avec perte.

Nous reconnaissons que l'approximation des petites chaînes n'est pas intéressante, car elle serait fort médiocre et l'intérêt du codage avec perte serait annihilé: pour les petites chaînes, nous avons besoin d'autant de bits, pour les coder avec perte qu'avec le codage CC. De plus, il nous faut préserver au mieux la forme des petites régions, qui correspondent généralement à des parties primordiales de la segmentation (yeux, bouches, oreilles, nez, etc.). C'est pourquoi nous allons utiliser pour les petits contours le codage sans perte CC décrit au paragraphe 3.3, et conserver la technique d'approximation pour les chaînes plus longues. D'après la Table 4.1, quatre classes de chaînes sont à prévoir. Le seuil correspondant à la longueur minimale de la chaîne à approximer, vaut ici 30. Il est évident que ce seuil est réglable en fonction des propriétés des cartes de segmentation. De plus, si nous rencontrons des courbes approximées, telles que la différence entre sommets vaut  $\Delta v \geq 32$  (pour les deux coordonnées), nous insérons un sommet pour rendre  $\Delta v < 32$ , mais cette situation est très rare. Pour coder la courbe polygonale, nous réservons 16 bits au premier sommet et nous

Classe	Type	Pels	$\Delta Vertex$	Bits
0	sans perte	$< 30$	-	$\sim 1.3$ /pel
1	avec perte	$> 30$	$0 \leq \Delta v_x, \Delta v_y < 8$	6 /vertex
2	avec perte	$> 30$	$8 \leq \Delta v_x, \Delta v_y < 16$	8 /vertex
3	avec perte	$> 30$	$16 \leq \Delta v_x, \Delta v_y < 32$	10 /vertex

TAB. 4.1 - Création de quatre classes de chaînes à coder

codons la différence des coordonnées des sommets consécutifs du polygone. Selon l'ensemble des distances entre ces coordonnées, nous associons à la chaîne une des

trois classes 1, 2 et 3 (2 bits) de codage avec perte.

### 4.3. Reconstruction

La procédure de reconstruction est très similaire à celle exposée dans la section 3.3.4. Pour les petites chaînes, nous appliquons exactement la même technique, puisqu'aucune approximation n'a été effectuée. Pour les contours approximatés, nous allons remplir l'intérieur des courbes polygonales obtenues. Pour réaliser un tel remplissage, il nous suffit de coder chaque pixel des arêtes du polygone par l'un des trois codes "S", "E" ou "I". L'ordre de remplissage est exactement le même que l'ordre de codage des différentes chaînes, comme pour le cas du codage sans perte CC décrit à la section 3.3.4.

### 4.4. Conclusion

Nous pourrions effectuer une étude plus poussée des propriétés de chaque chaîne pour une meilleure définition des classes, et ensuite appliquer les codages associés aux classes. En observant les résultats, nous concluons que cette méthode d'approximation est très bonne. Le taux de distorsion peut être contrôlé et réduit. Evidemment, plus le débit de transmission est diminué, plus les phénomènes de distorsion apparaissent sur l'image reconstruite. Au lieu d'une approximation polygonale, nous aurions pu appliquer des fonctions B-splines [46], ce qui aurait légèrement amélioré l'allure des courbes. D'après les résultats du Tableau 4.2 et de la Fig. 4.2, un taux de distorsion (quadratique) proche de 1 est toléré. Ceci laisse une grande marge d'action sur l'approximation. Dans le cas de la séquence "Miss America", les taux de bits par pixel sont maintenant inférieurs à ceux fixés par un débit de 16 kbits/s (0.06 bit/pel).



	$\gamma = 7$			$\gamma = 10$			$\gamma = 13$		
	$\delta(\Delta)$	bits	bit/pel	$\delta(\Delta)$	bits	bit/pel	$\delta(\Delta)$	bits	bit/pel
(M)	0.55	1223	0.048	1.1	991	0.039	1.54	879	0.035
(C)	0.29	4044	0.160	0.54	3268	0.129	0.84	2805	0.111
(F)	0.35	4662	0.184	0.74	3718	0.147	1.05	3254	0.128

TAB. 4.2 - Codage avec perte de “Miss America” #0, “Carphone” #0 et “Foreman” #0 pour la segmentation intra-trame

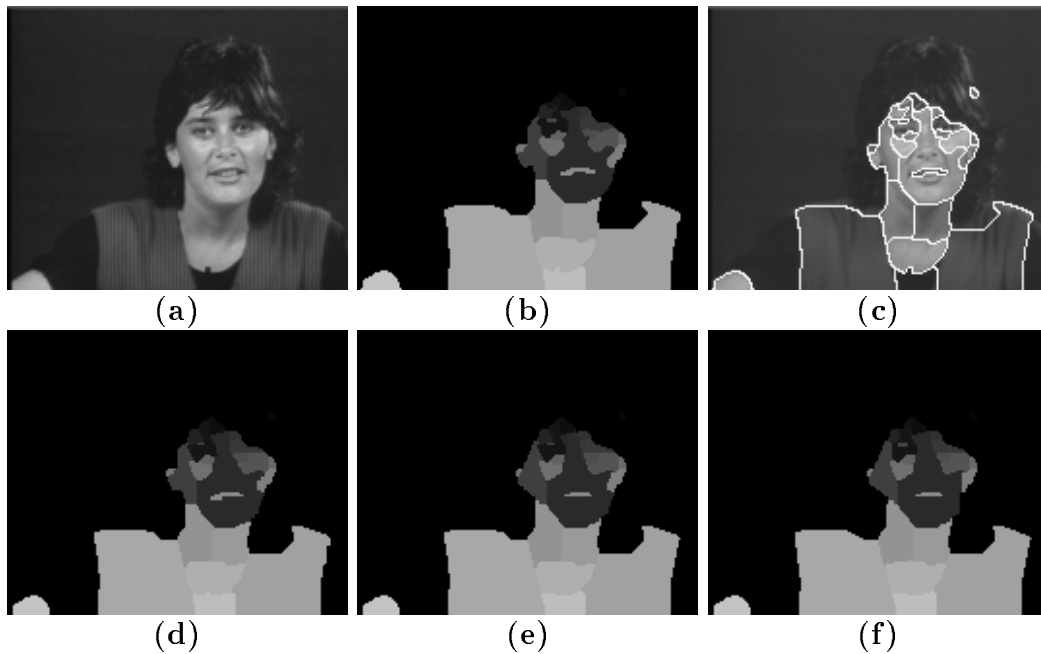


FIG. 4.2 - “Miss America” #90: (a) trame originale, (b) segmentation intra, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs d’approximation (d)  $\gamma = 7$ , (e)  $\gamma = 10$  et (f)  $\gamma = 13$



# Chapitre 5

## Résultats pour les séquences d'images

### 5.1. Outils de simulation disponibles au sein du groupe vidéo de l'INRS

Avant de présenter les images et tableaux de résultats, il m'apparaît nécessaire de mentionner quelques outils du groupe vidéo de l'INRS, qui m'ont été de première utilité pendant le projet de recherche. Nous avons implanté les programmes de simulation dans le langage C++. Les images vidéo ont été visualisées par le logiciel XView. Les simulations ont été réalisées sur les stations DECstation 5000. Pour la visualisation des séquences d'images, nous avons utilisé le système "Viewstore" de Viewgraphics.

### 5.2. Images de référence et types de segmentation

Afin de réaliser la simulation du codage des cartes de segmentation, nous avons utilisé trois séquences vidéo au format QCIF ( $176 \times 144$ ): "Miss America" (M),

“Carphone” (C) et “Foreman” (F) (Fig. 5.1). Deux segmentations ont donc été effectuées (cf section 2.3.1); une segmentation intra-trame basée-luminance (ex: Fig. 5.2) et une segmentation inter-trame basée-mouvement (ex: Fig. 5.16). La seconde fut réalisée par l’utilisation d’une segmentation initiale basée-luminance moyennement fine, combinée aux paramètres de mouvement. En vue d’une bonne estimation des paramètres de mouvement, les bords des trames ont été masqués. Nous présentons les résultats pour ces deux types de segmentation.

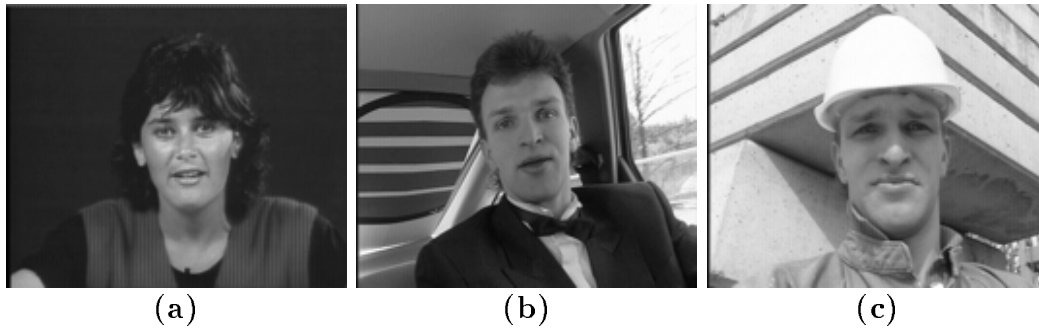


FIG. 5.1 - Séquences de référence (a) “Miss America”, (b) “Carphone” et (c) “Foreman”

### 5.3. Résultats pour la segmentation intra-trame

La séquence de référence “Miss America” (150 trames) a été sous-échantillonnée toutes les cinq trames, pour produire 30 trames segmentées. Les deux autres séquences, “Carphone” (382 trames) et “Foreman” (400 trames) ont aussi été partiellement traitées. Finalement, les séquences pour le traitement sont constituées des trames:

- “Miss America”: 0, 5, 10,  $\dots$ , 145,
- “Carphone”: 0, 1, 2,  $\dots$ , 19, 120, 123,  $\dots$ , 177,
- “Foreman”: 0, 1, 2,  $\dots$ , 19, 150, 153,  $\dots$ , 207.

Le Tableau 5.1 présente le nombre moyen de régions et le nombre moyen de points de contour, respectivement calculés sur les trois séquences traitées en mode intra.

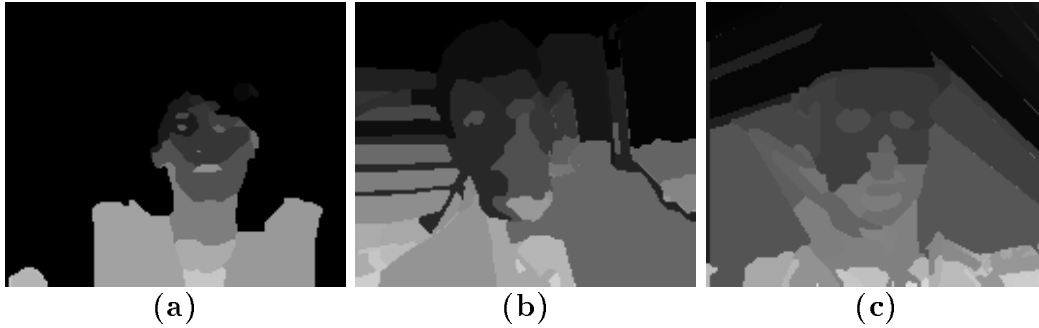


FIG. 5.2 - Exemple de segmentations intra-trame de (a) “Miss America” #105, (b) “Carphone” #123, et (c) “Foreman” #180 obtenues par la méthode basée sur MDL

	régions	cp
“Miss America”	20	830
“Carphone”	55	2425
“Foreman”	75	2760

TAB. 5.1 - Nombre moyen de régions et nombre moyen des points de contour (cp) pour les trois séquences traitées en mode intra

### 5.3.1. Codage sans perte

Définissons le taux comparatif de bits par point de contour (noté par  $b/cp$ ) par

$$b/cp = \frac{\text{Taille du bitstream final à transmettre}}{\text{Nombre de points de contour}}$$

Chaque point de contour de cartes de segmentation a été compté une fois et une seule. Par la suite, le nombre de bits par pixel d’une trame sera noté par  $b/p$ .

D’après le Tableau 5.2, dans lequel les valeurs représentent une moyenne sur chaque séquence, nous obtenons un meilleur taux de compression avec CC utilisant un codage conditionnel. Remarquons dans ce cas que le nombre total de bits pour coder les images segmentées est donné. Cependant, pour calculer le taux comparatif

	RLC			BAC			CC			
	bits	b/p	b/cp	bits	b/p	b/cp	$\hat{H}_{co}$	bits	b/p	b/cp*
(M)	4000	.158	4.8	2135	.084	2.57	1.20	1447	.057	1.36
(C)	7230	.285	2.98	5885	.232	2.43	1.16	4010	.158	1.29
(F)	9325	.368	3.38	8150	.322	2.95	1.24	5090	.201	1.41

TAB. 5.2 - Codage sans perte des segmentations intra-trame. (\*) = sans compter le codage des points de départ des chaînes

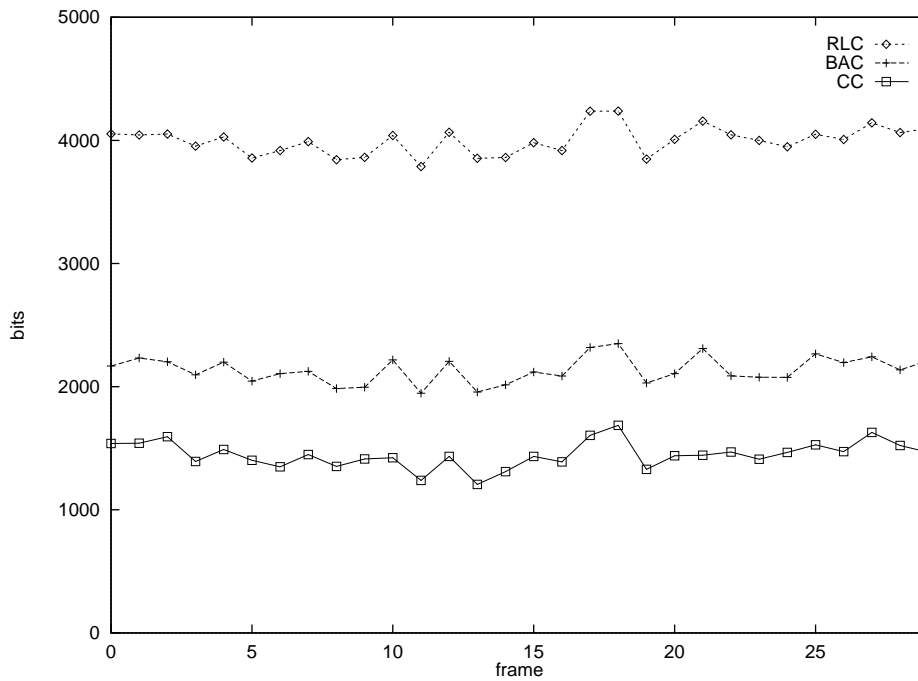


FIG. 5.3 - Nombre de bits par intra-trame pour la séquence "Miss America"

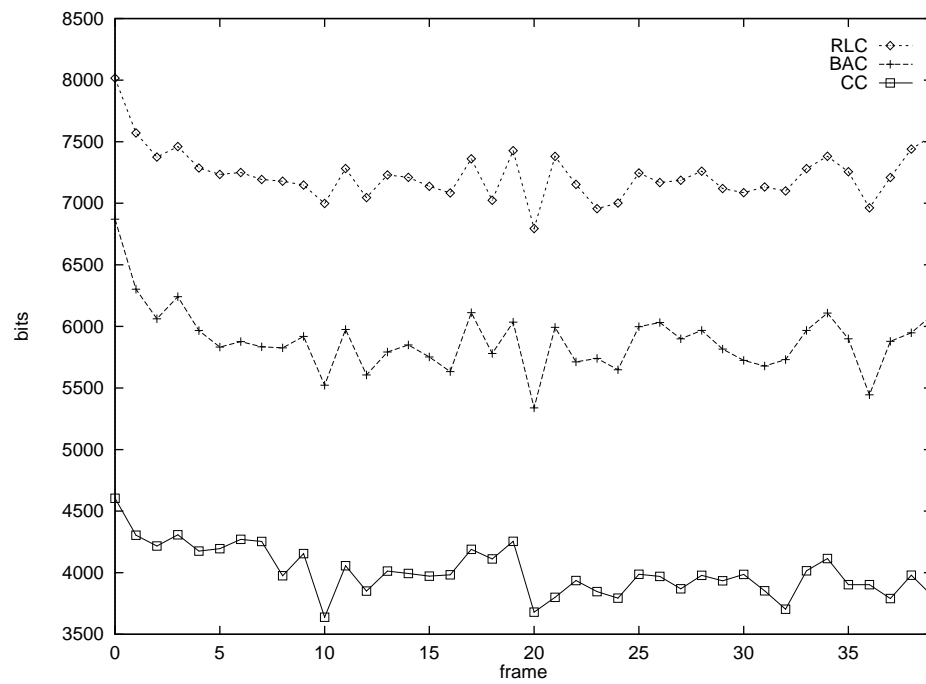


FIG. 5.4 - Nombre de bits par intra-trame pour la séquence "Carphone"

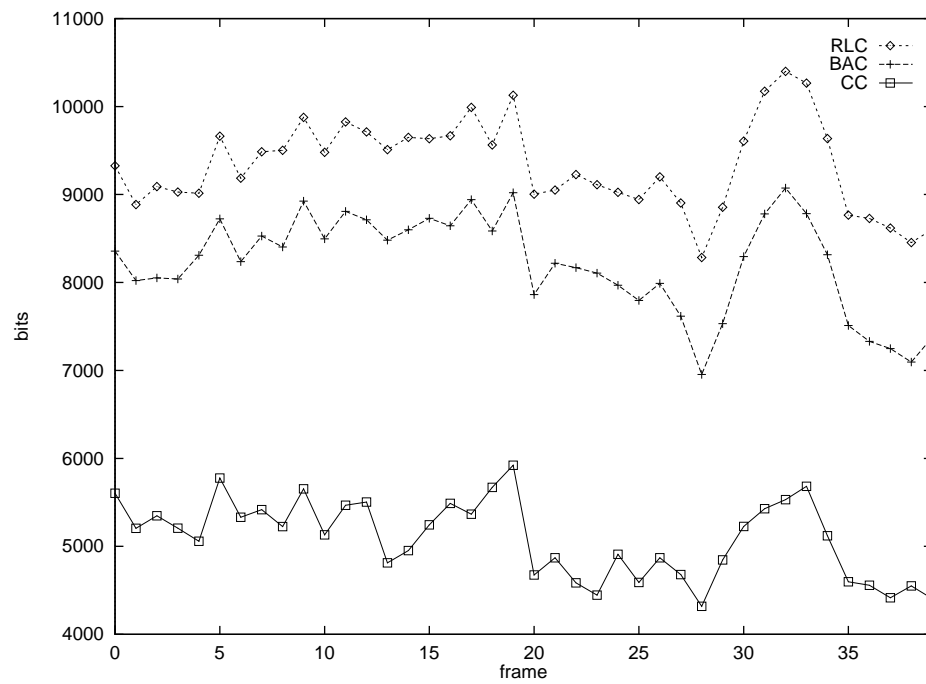


FIG. 5.5 - Nombre de bits par intra-trame pour la séquence "Foreman"

$b/cp$ , nous lui avons soustrait les bits réservés au codage des points de départ des chaînes, ceci pour pouvoir comparer les différents taux. Bien entendu, l'efficacité de CC dépend de la valeur de de l'entropie  $\hat{H}_{co}$  de la source. Ici, l'écart entre  $\hat{H}_{co}$  et  $\hat{L}_{co}$  n'excède pas 20%, ce qui est acceptable. Tous les résultats de codage sans perte des segmentations intra, avec les trois méthodes RLC, BAC, CC, sont présentés aux Fig. 5.3, 5.4 et 5.5.

### 5.3.2. Codage avec perte

	$\gamma = 7$				$\gamma = 10$				$\gamma = 13$			
	$\delta(\Delta)$	bits	b/p	b/cp	$\delta(\Delta)$	bits	b/p	b/cp	$\delta(\Delta)$	bits	b/p	b/cp
(M)	0.47	1234	.049	1.48	0.95	1012	.040	1.22	1.45	900	.036	1.08
(C)	0.31	3485	.138	1.44	0.61	2805	.111	1.16	0.91	2440	.096	1.01
(F)	0.36	4257	.168	1.54	0.72	3392	.133	1.23	1.1	2940	.116	1.06

TAB. 5.3 - Codage avec perte des segmentations intra-trame

D'après le Tableau 5.3 (de même, les valeurs représentent une moyenne sur la séquence), nous pouvons augmenter le taux de distorsion  $\delta(\Delta)$  défini en section 4.1.3. De cette manière, le taux comparatif peut décroître vers 1, voir en-dessous, en augmentant le facteur d'approximation et donc le taux de distorsion. Cependant, il est préférable de ne pas trop faire croître le taux de distorsion, car sinon, les résultats visuels vont présenter de nombreux défauts qui ne seront plus acceptables. Les artéfacts peuvent être observés aux Fig. 5.13, 5.14 et 5.15, généralement aux régions approximées des épaules, de la tête (oreilles, bouche, yeux...). Nous pensons qu'un taux de distorsion, ne dépassant guère la valeur 1, est acceptable pour un bon codage avec perte. Les Fig. 5.6, 5.8, 5.10 présentent les nombres de bits par intra-trame pour les trois séquences. Les mesures de distorsion sont données aux Fig. 5.7, 5.9, 5.11. Remarquez que les variations du nombre de bits entre chaque trame, non reflétées dans les moyennes, sont assez importantes; la compression peut être bien meilleure sur une trame que sur une autre d'une même séquence. A la Fig.



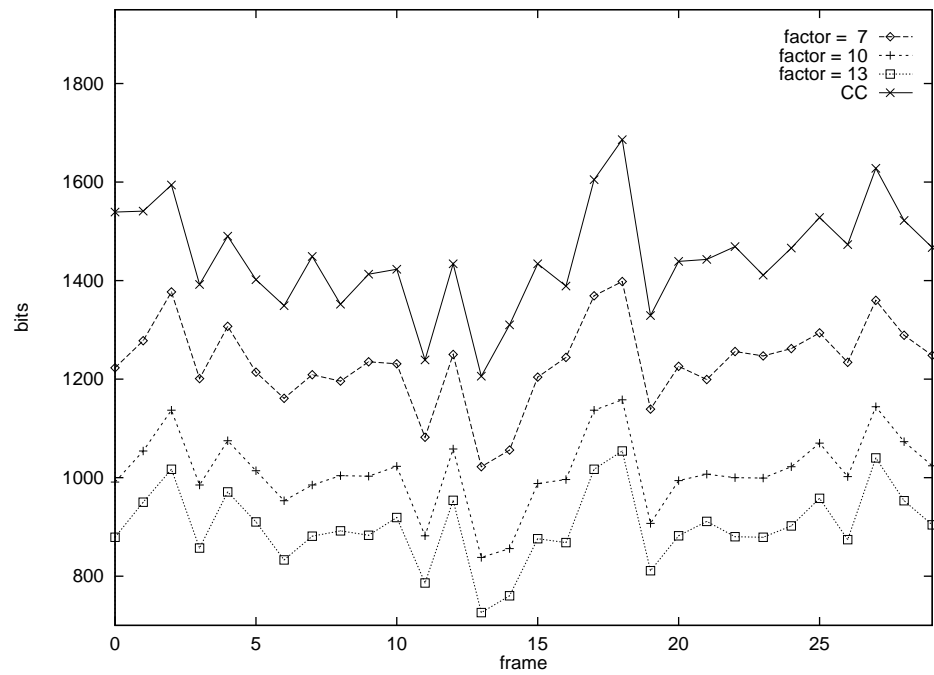


FIG. 5.6 - Nombre de bits par intra-frame pour la séquence “Miss America”, avec les facteurs  $\gamma = 7, 10, 13$

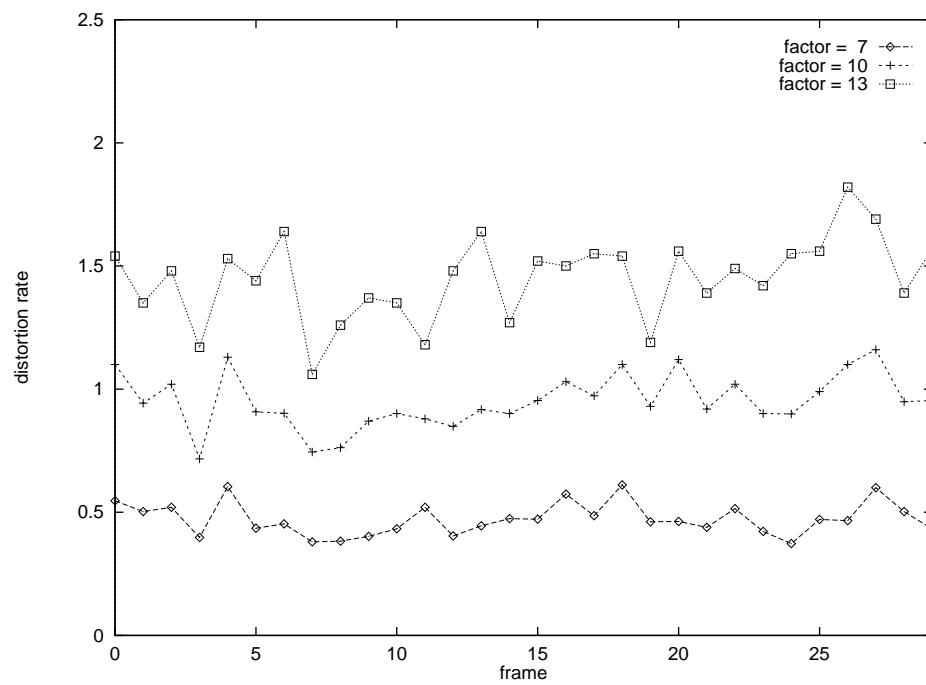


FIG. 5.7 - Mesure de distorsion intra-frame pour la séquence “Miss America”, avec les facteurs  $\gamma = 7, 10, 13$

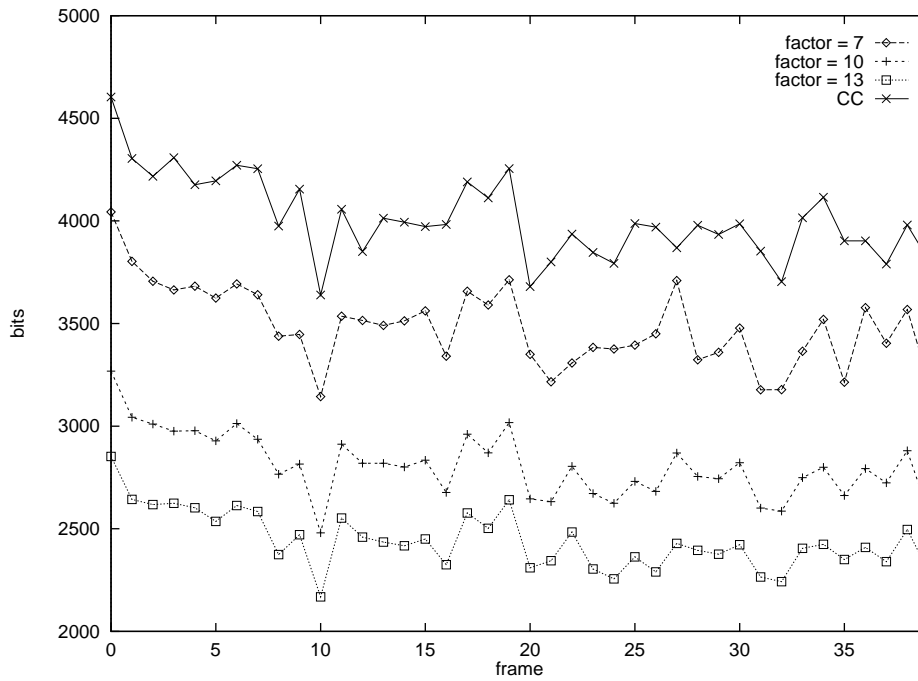


FIG. 5.8 - Nombre de bits par intra-frame pour la séquence “Carphone”, avec les facteurs  $\gamma = 7, 10, 13$

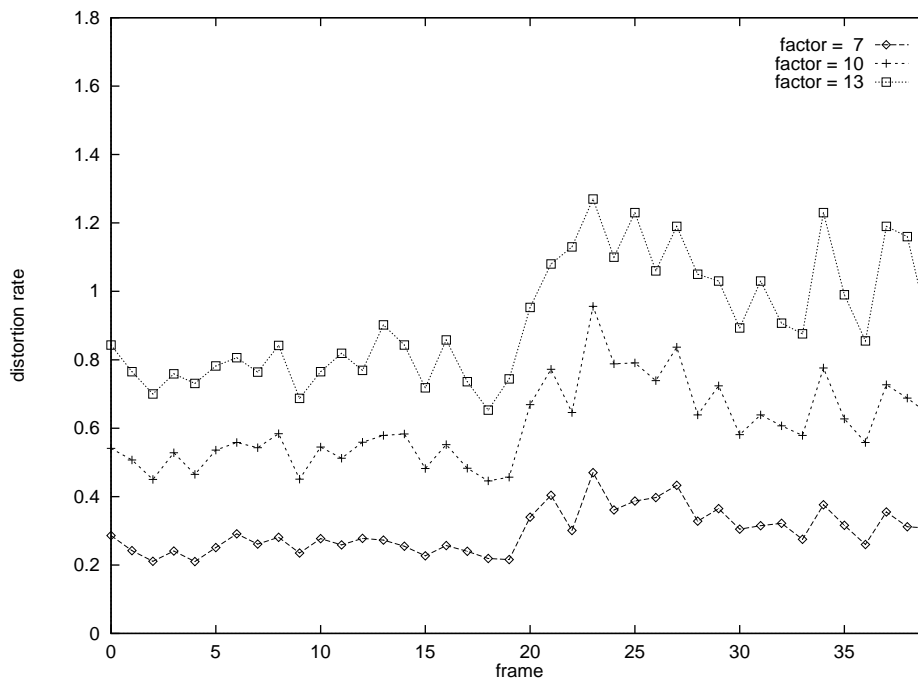


FIG. 5.9 - Mesure de distorsion par intra-frame pour la séquence “Carphone”, avec les facteurs  $\gamma = 7, 10, 13$

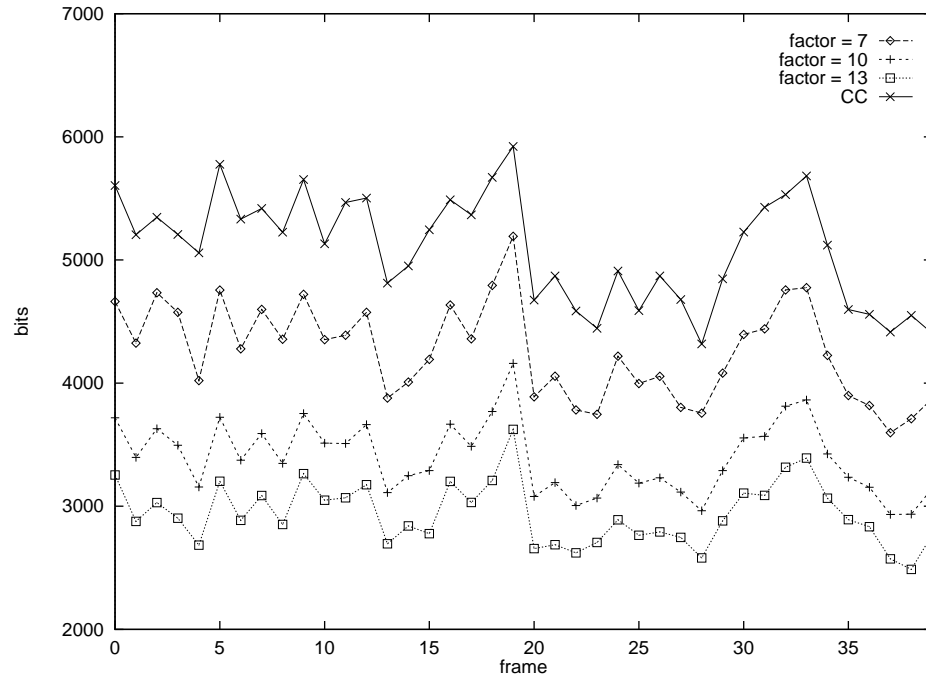


FIG. 5.10 - Nombre de bits par intra-trame pour la séquence “Foreman”, avec les facteurs  $\gamma = 7, 10, 13$

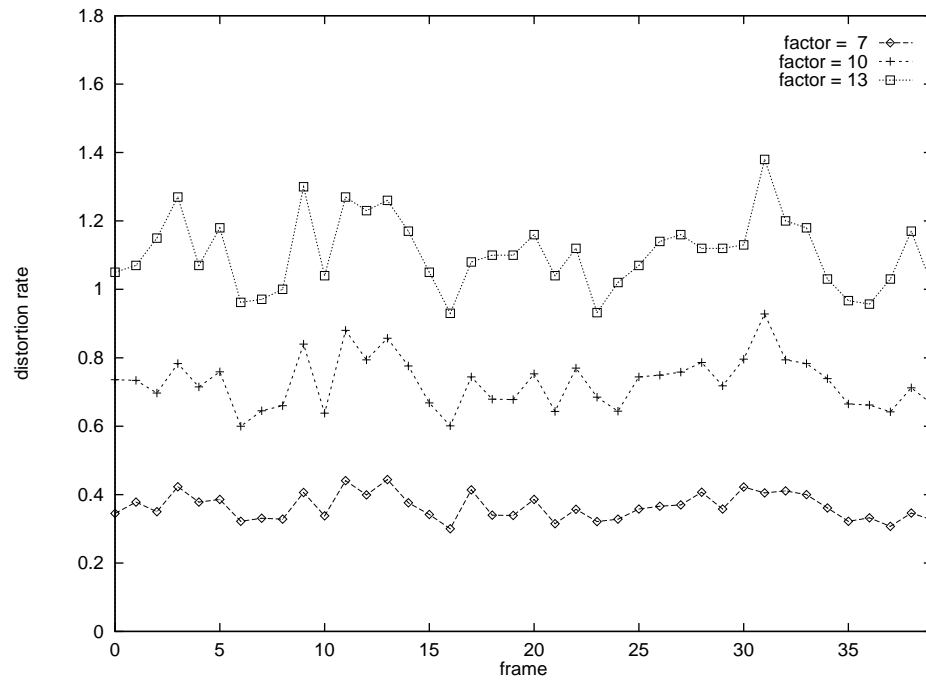


FIG. 5.11 - Mesure de distorsion par intra-trame pour la séquence “Foreman”, avec les facteurs  $\gamma = 7, 10, 13$

5.12, nous avons représenté les valeurs trouvées sous forme de nuage de données. Cette représentation est très significative; pour chaque séquence, le nuage suit les caractéristiques classiques du compromis distorsion-débit.

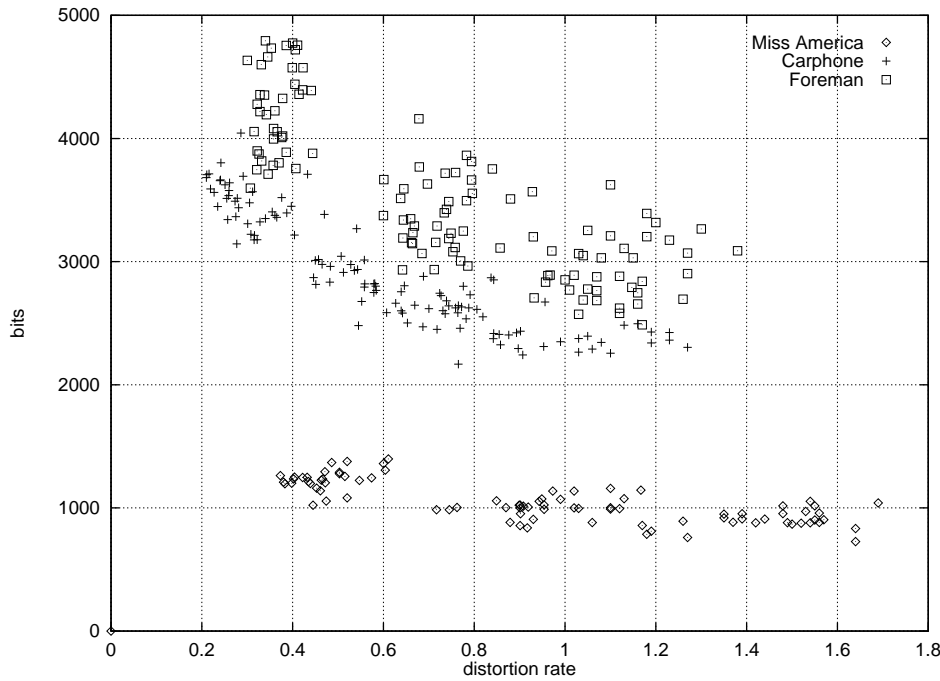


FIG. 5.12 - Evolution des bits avec la mesure de distorsion pour les trois séquences

## 5.4. Résultats pour la segmentation inter-trame

Les séquences pour le traitement sont constituées des trames:

- "Miss America": 3, 6, 9,  $\dots$ , 57,
- "Carphone": 120, 123,  $\dots$ , 177,
- "Foreman": 150, 153,  $\dots$ , 207.

Le Tableau 5.4 présente le nombre moyen de régions et le nombre moyen de points de contour, respectivement calculés sur les trois séquences traitées en mode inter.

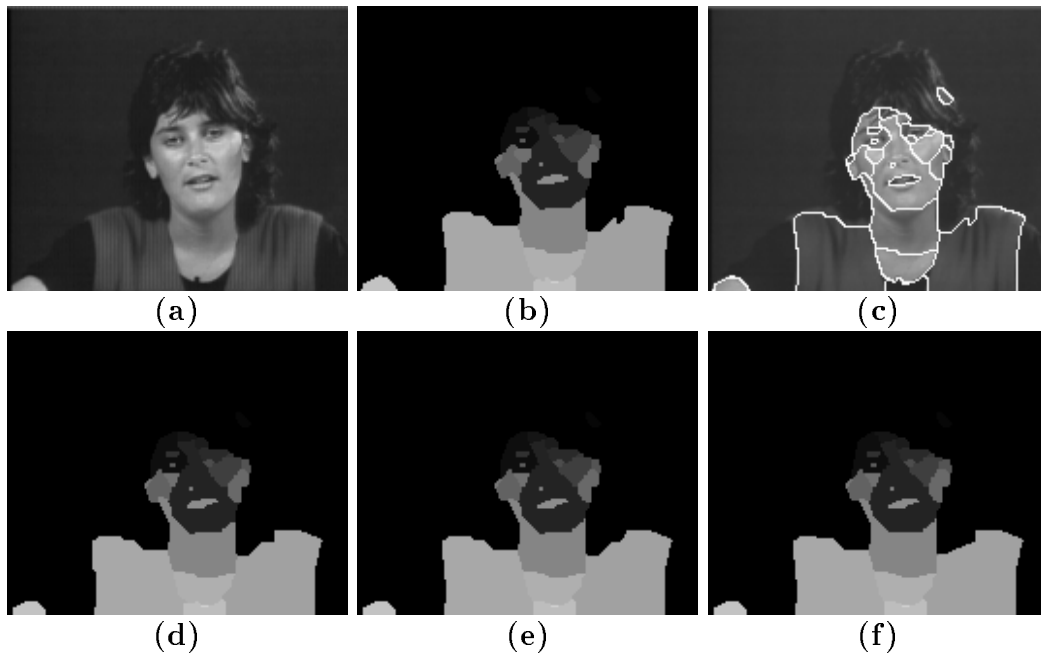


FIG. 5.13 - "Miss America" #135: (a) trame originale, (b) segmentation intra, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d)  $\gamma = 7$ , (e)  $\gamma = 10$  et (f)  $\gamma = 13$

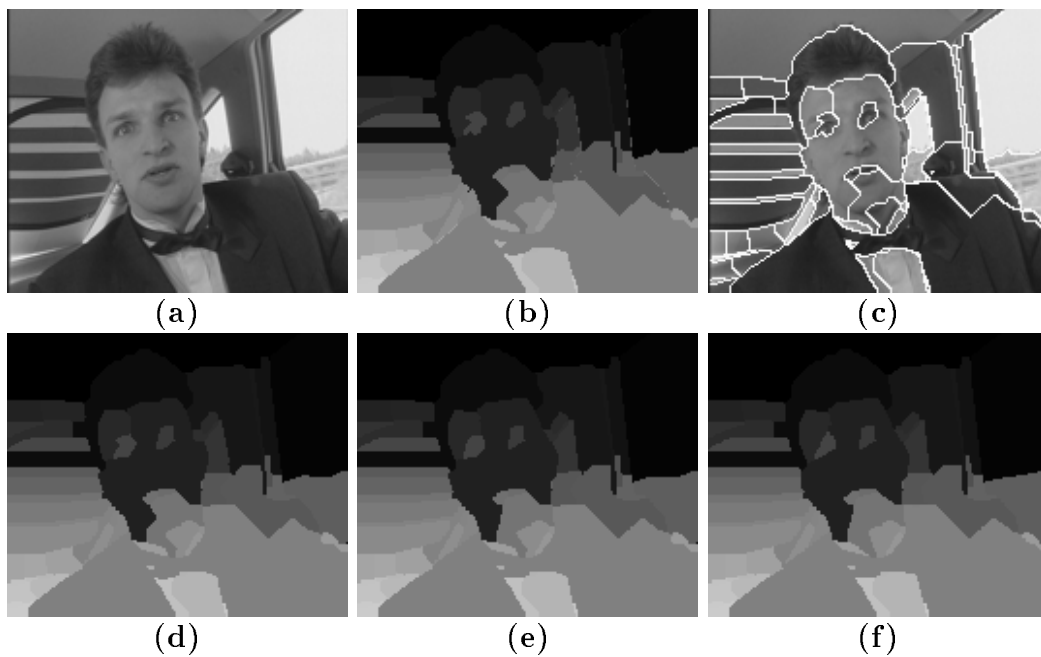


FIG. 5.14 - "Carphone" #159: (a) trame originale, (b) segmentation intra, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d)  $\gamma = 7$ , (e)  $\gamma = 10$  et (f)  $\gamma = 13$

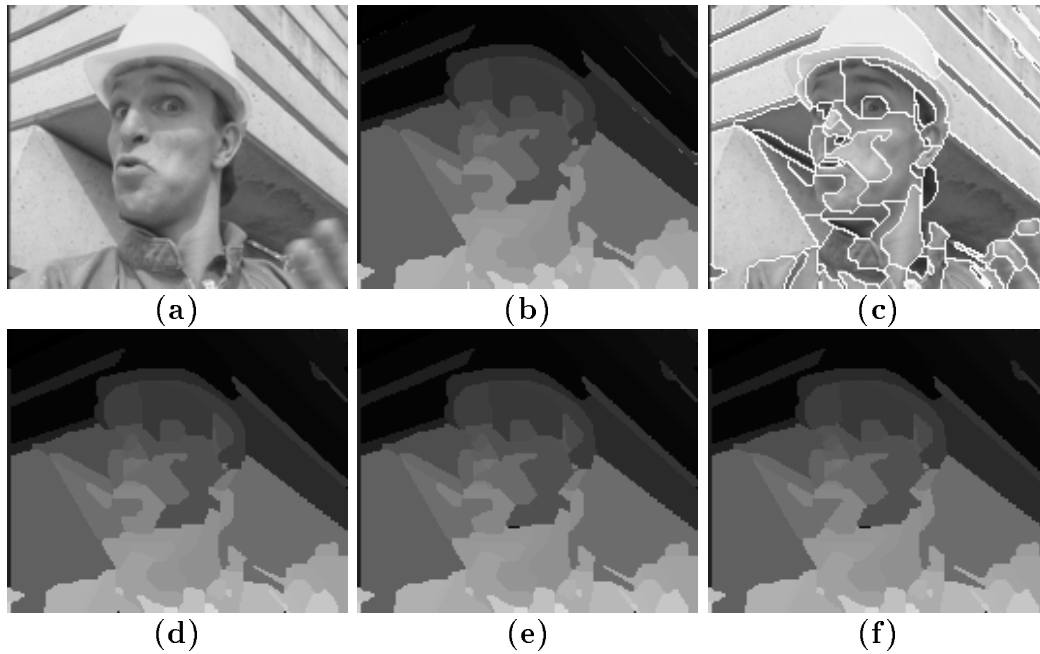


FIG. 5.15 - “Foreman” #189: (a) frame originale, (b) segmentation intra, (c) contours de segmentation sur la frame originale; reconstruction avec perte pour les facteurs (d)  $\gamma = 7$ , (e)  $\gamma = 10$  et (f)  $\gamma = 13$

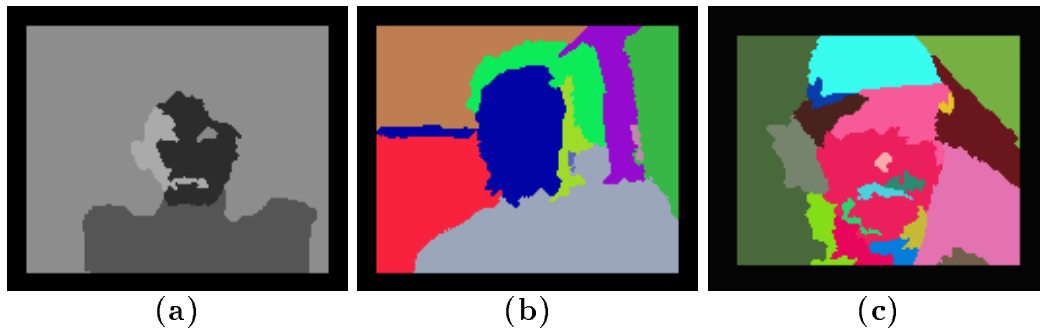


FIG. 5.16 - Exemple de segmentations inter-trame de (a) “Miss America” #6, (b) “Carphone” #3, et (c) “Foreman” #150 obtenues par la procédure “basé-mouvement”

	régions	cp
“Miss America”	5	540
“Carphone”	10	875
“Foreman”	15	1115

TAB. 5.4 - Nombre moyen de régions et nombre moyen des points de contour (cp) pour les trois séquences traitées en mode inter

### 5.4.1. Codage sans perte

	RLC			BAC			CC			
	bits	b/p	b/cp	bits	b/p	b/cp	$\hat{H}_{co}$	bits	b/p	b/cp*
(M)	2622	.103	4.85	1380	.054	2.55	1.07	767	.030	1.27
(C)	2851	.112	3.26	2200	.087	2.51	1.16	1380	.054	1.39
(F)	4685	.185	4.20	3081	.122	2.76	1.17	1810	.071	1.40

TAB. 5.5 - Codage sans perte des segmentations inter-trame. (\*) = sans compter le codage des points de départ des chaînes

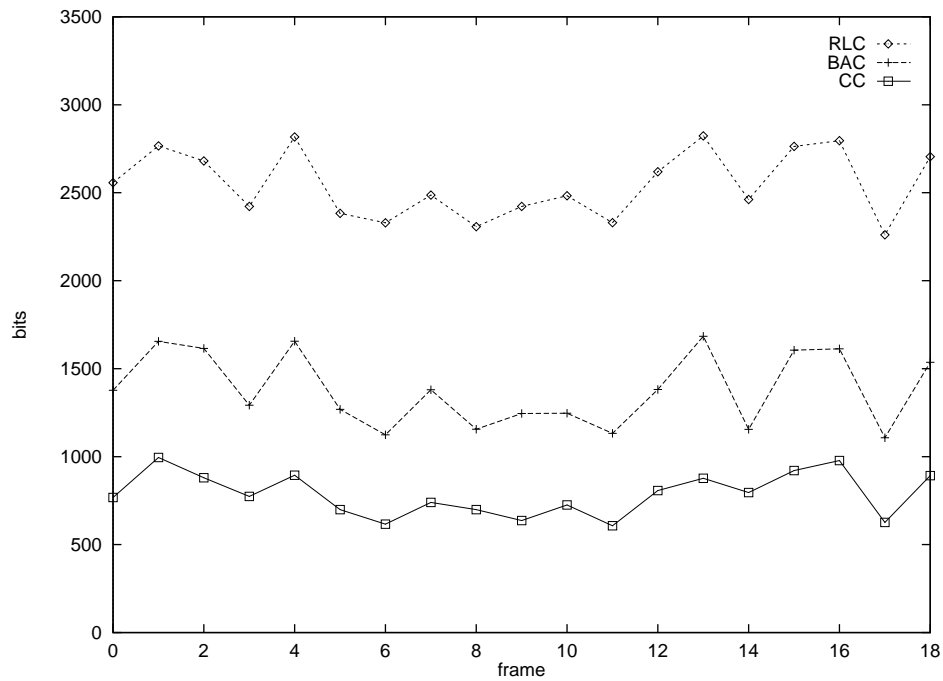


FIG. 5.17 - Nombre de bits par inter-trame pour la séquence "Miss America"

De même que pour les résultats concernant les segmentations intra-trame, le meilleur codage sans perte est obtenu par la technique CC. Comme les valeurs  $\hat{H}_{co}$  sont plus petites, les taux  $b/cp$  sont légèrement réduits, mais restent cependant dans les mêmes ordres de grandeur. Par contre, comme le nombre de points de contour a été énormément réduit par la procédure de segmentation inter-trame, de ce fait, il nous faut moins de bits pour véhiculer l'information de ces cartes de

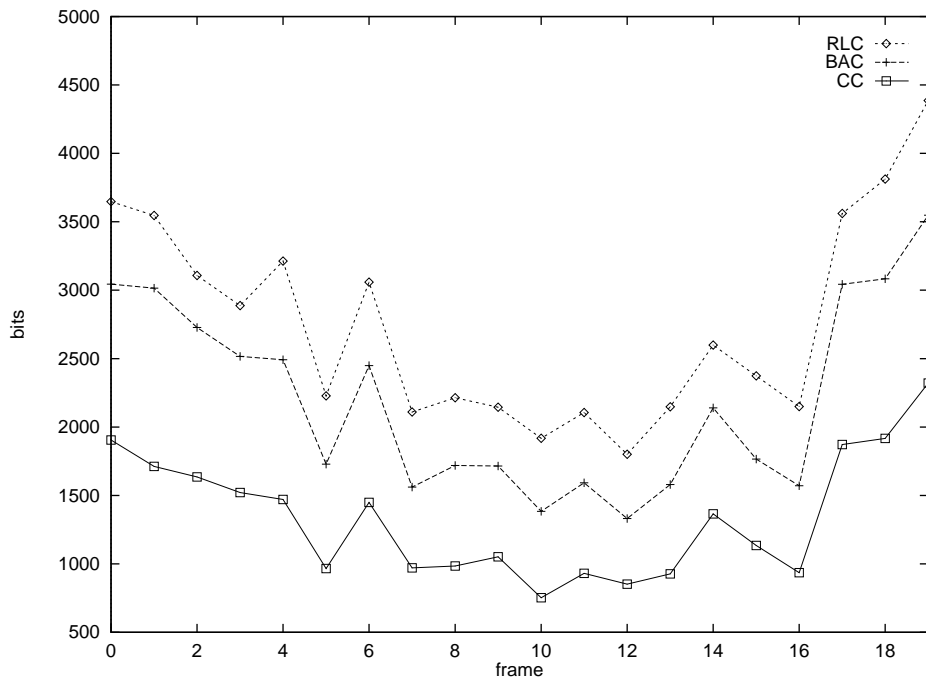


FIG. 5.18 - Nombre de bits par inter-trame pour la séquence "Carphone"

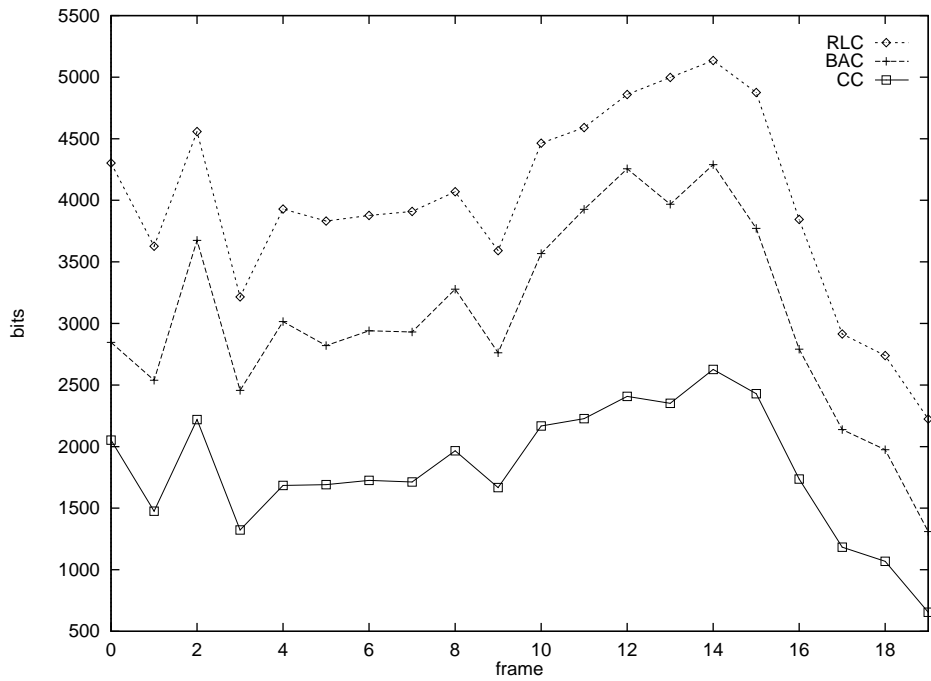


FIG. 5.19 - Nombre de bits par inter-trame pour la séquence "Foreman"



segmentation. Tous les résultats de codage sans perte des segmentations inter, avec les trois méthodes RLC, BAC, CC, sont présentés aux Fig. 5.17, 5.18 et 5.19.

### 5.4.2. Codage avec perte

	$\gamma = 10$				$\gamma = 13$				$\gamma = 16$			
	$\delta(\Delta)$	bits	b/p	b/cp	$\delta(\Delta)$	bits	b/p	b/cp	$\delta(\Delta)$	bits	b/p	b/cp
(M)	0.69	482	.019	0.89	1.05	383	.015	0.71	1.43	331	.013	0.62
(C)	1.75	831	.033	0.95	2.89	725	.029	0.83	3.92	635	.025	0.72
(F)	1.28	1285	.051	1.15	2.1	1140	.045	1.02	2.88	995	.039	0.89

TAB. 5.6 - Codage avec perte des segmentations inter-frame

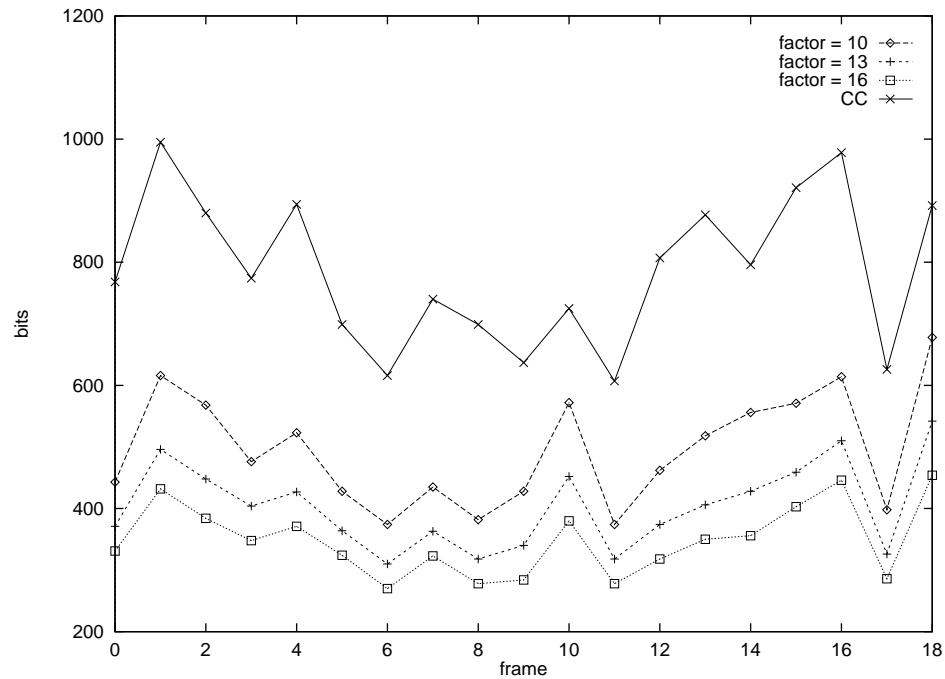


FIG. 5.20 - Nombre de bits par inter-frame pour la séquence "Miss America", avec les facteurs  $\gamma = 7, 10, 13$

D'après les derniers résultats présentés dans les tableaux 5.5 et 5.6 (moyennes sur chaque séquence), il nous faut évidemment bien moins de bits pour coder l'information des cartes de segmentation inter-frame. Ceci s'explique par le nombre

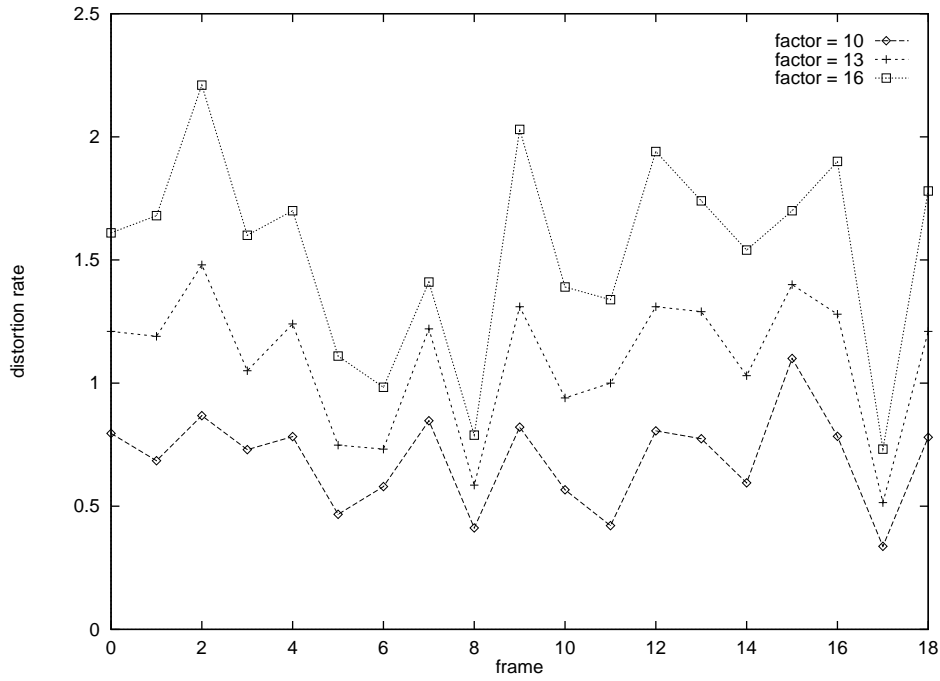


FIG. 5.21 - Taux de distortion par inter-frame pour la séquence "Miss America", avec les facteurs  $\gamma = 7, 10, 13$

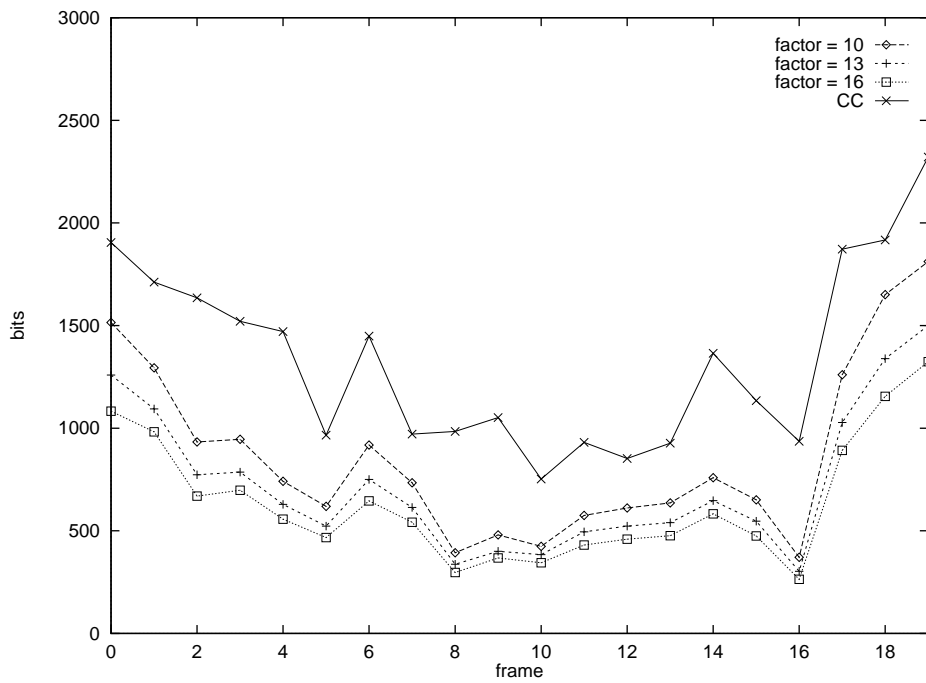


FIG. 5.22 - Mesure de bits par inter-frame pour la séquence "Carphone", avec les facteurs  $\gamma = 7, 10, 13$

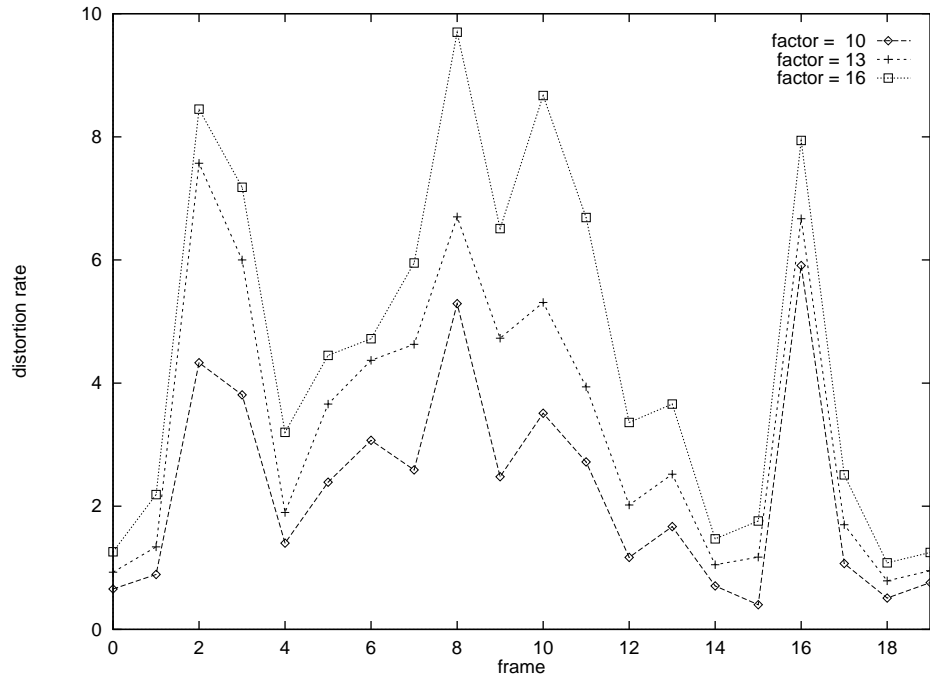


FIG. 5.23 - Mesure de distortion par inter-frame pour la séquence “Carphone”, avec les facteurs  $\gamma = 7, 10, 13$

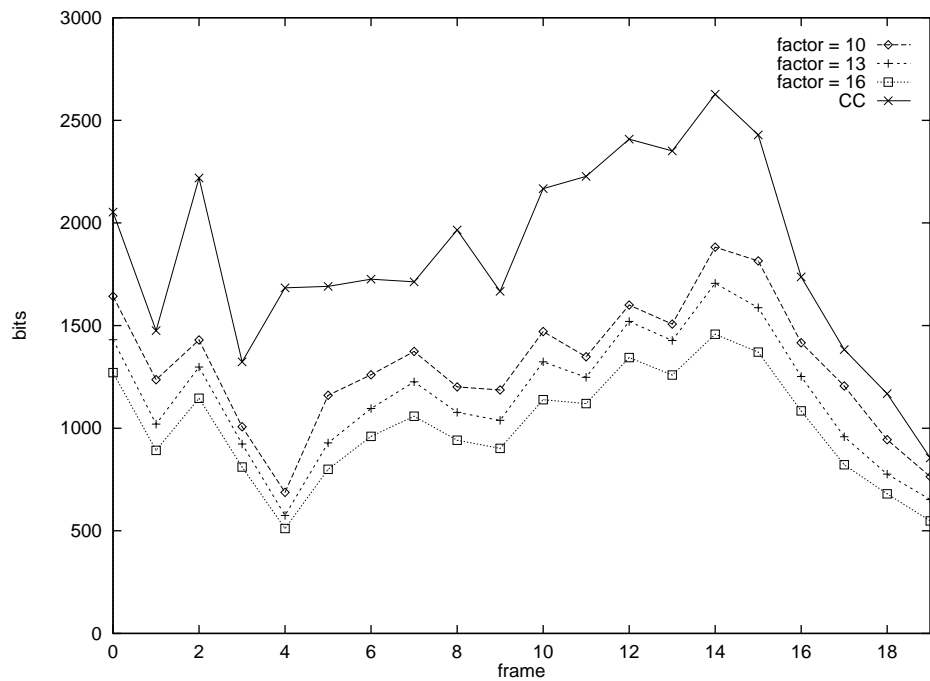


FIG. 5.24 - Nombre de bits par inter-frame pour la séquence “Foreman”, avec les facteurs  $\gamma = 7, 10, 13$

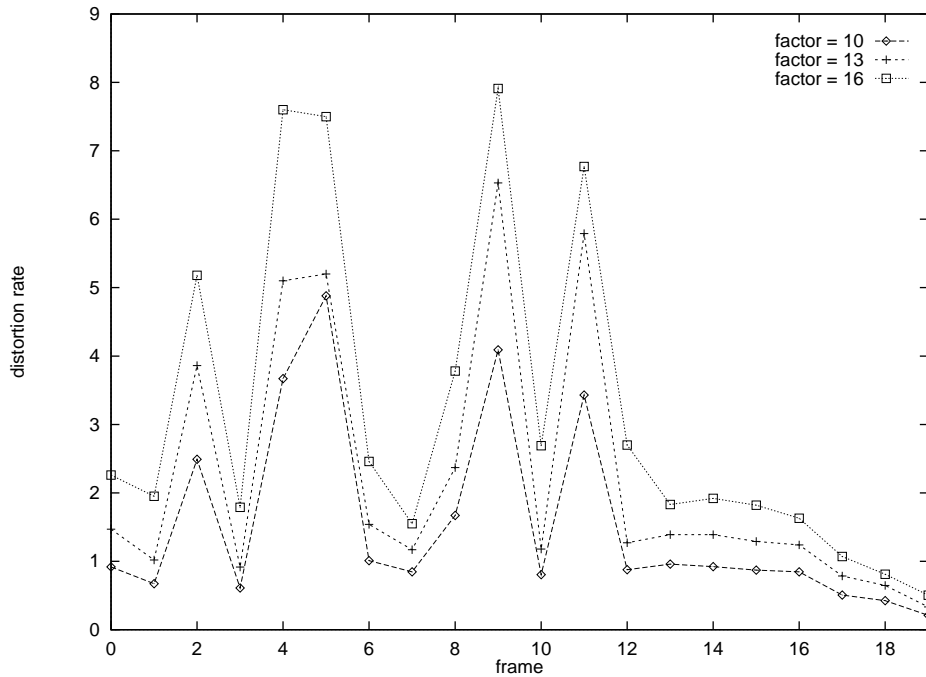


FIG. 5.25 - Mesure de distorsion par inter-trame pour la séquence “Foreman”, avec les facteurs  $\gamma = 7, 10, 13$

réduit de régions par trame, donc par moins de points de contour à coder. Toutefois, il nous est difficile de donner un seuil de taux de distorsion à ne pas dépasser. En fonction de chaque séquence, contenant des images segmentées de nature distincte, les taux de distorsion varient beaucoup. Cependant, il est prévisible qu’il faut garder ce taux dans des grandeurs raisonnables, sachant que la compression est de toute façon améliorée par le codage avec perte. Les Fig. 5.20, 5.22, 5.24 présentent les nombres de bits par inter-trame pour les trois séquences. Les mesures de distorsion sont données aux Fig. 5.21, 5.23, 5.25. Nous remarquons que les variations de la distorsion entre chaque trame est souvent importante, surtout pour “Carphone” et “Foreman”. Ceci est dû à la difficulté d’obtenir de bonnes segmentations en mode inter, car dans ces séquences, le comportement du mouvement est très complexe. Il en découle des approximations de nature différente pour chaque trame. En conséquence, le nuage de la Fig. 5.26 fait légèrement moins bien ressortir la corrélation distorsion-

débit.

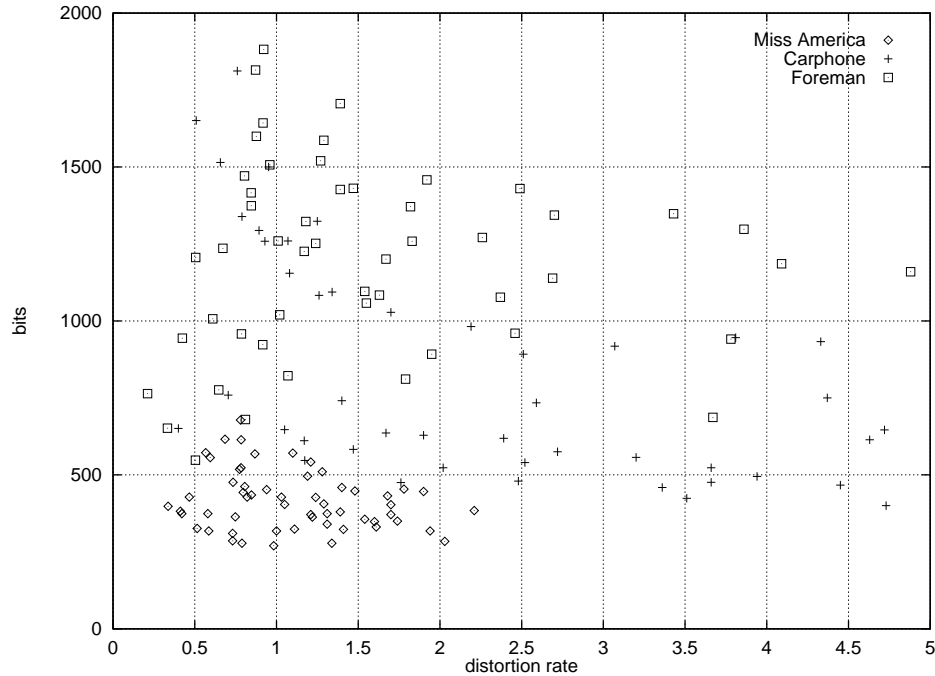


FIG. 5.26 - Evolution des bits avec le taux de distorsion pour les trois séquences

## 5.5. Commentaires

Pour le codage des cartes de segmentation, nous avons finalement deux solutions concrètes. Si nous choisissons d'utiliser un codage sans perte, il est indispensable de considérer la méthode CC. Les résultats présentés pour ce codage sont intéressants, bien que la quantité de bits représentative de l'information des points de départ des chaînes reste excessive. Pour la séquence "Miss America", il nous faut au plus 1500 bits pour coder une segmentation intra-trame, et au plus 800 bits pour coder celle inter-trame. Ces résultats sont relativement concluants.

Au cours du mémoire, nous avons tenté de mettre en valeur une technique de codage avec perte, qui semble bien appropriée pour le codage des cartes de

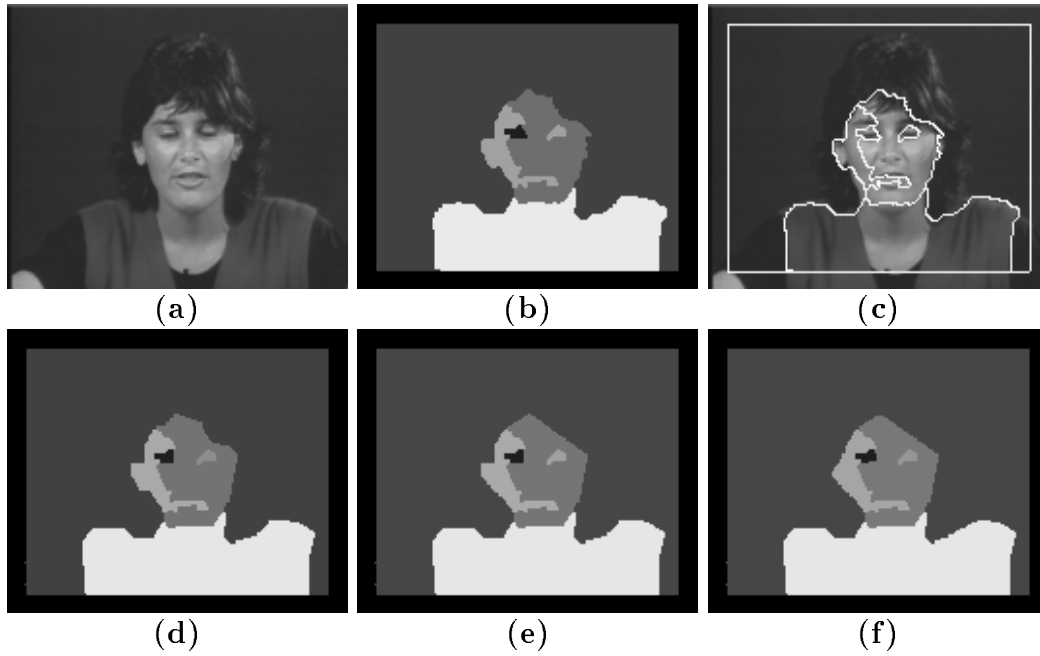


FIG. 5.27 - “Miss America” #6: (a) trame originale, (b) segmentation inter, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d) 10, (e) 13 et (f) 16

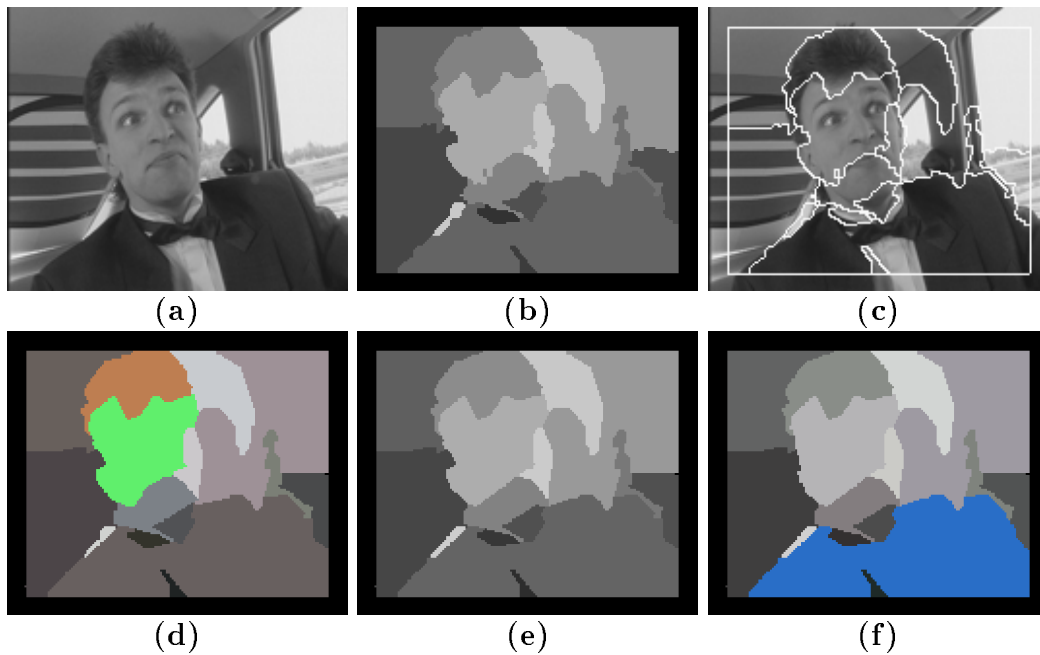


FIG. 5.28 - “Carphone” #120: (a) trame originale, (b) segmentation inter, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d) 10, (e) 13 et (f) 16

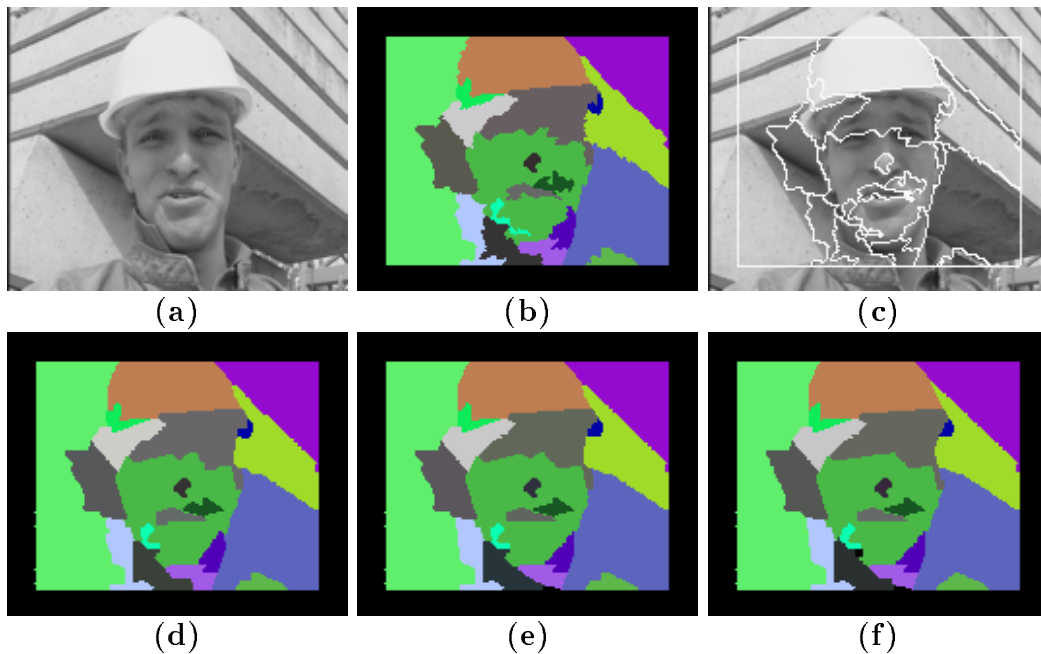


FIG. 5.29 - “Foreman” #150: (a) trame originale, (b) segmentation inter, (c) contours de segmentation sur la trame originale; reconstruction avec perte pour les facteurs (d) 10, (e) 13 et (f) 16

segmentation; il nous est possible de régler le nombre de vertex par chaîne à approximer ou bien le taux de distorsion prédéfini. Or, si nous forçons la mesure de distorsion à rester assez petite, la reconstruction des cartes de segmentation garde une qualité visuelle subjective encourageante. Cette remarque nous autorise à envisager d'utiliser cette méthode pour coder la plupart des cartes de segmentation rencontrées dans le diagramme de la Fig. 2.2. Aussi, pour la séquence “Miss America”, avec un taux de distorsion (quadratique) d'ordre de grandeur 1, il nous faut désormais au plus 1000 bits pour coder une segmentation intra-trame, et au plus 400 bits pour coder celle inter-trame.





# Chapitre 6

## Conclusion

Ce mémoire traite du problème de la compression des cartes de segmentation dans les séquences d'images. La recherche a été effectuée dans le cadre d'une activité du groupe des Communications Visuelles de l'INRS portant sur le codage vidéo à très bas débit. Plus précisément, une étude comparative de diverses techniques a été menée, dans le but d'obtenir le meilleur taux de compression pour coder les cartes de segmentation. Plusieurs méthodes de codage sans perte et avec perte ont été testées, au travers de simulations effectuées sur des séquences de référence. Les résultats ont été présentés dans le chapitre précédent. Ils démontrent que les techniques étudiées ont des efficacités inégales. Le meilleur codage sans perte reste le codage CC, qui permet une compression intéressante de l'information de segmentation, environ égal à 1.3 bits par point de contour dans les simulations. En fait, ce taux dépend énormément de la nature de l'image segmentée; le résultat de la segmentation va déterminer la distribution des directions conditionnelles, soit l'entropie de source, et donc la longueur moyenne du code. D'après les estimations faites sur ces deux entités, la longueur moyenne du code utilisé reste proche de notre estimation de l'entropie conditionnelle (ou taux d'entropie  $H_\tau$ ). L'écart, inférieur à 20% renforce la supériorité du codage sans perte CC pour les cartes de segmentation.

Un autre point important, proposé dans cette thèse, réside dans l'application aux cartes de segmentation d'un codage efficace avec perte. En effectuant une approximation polygonale optimale sur les contours de région, le taux de compression est amélioré, et il est réduit selon une loi inversement proportionnelle à la mesure de distorsion de l'approximation. Les résultats visuels sont subjectivement satisfaisants et garants de l'efficacité de ce codage avec perte. Mieux même, il nous est possible de contrôler le taux de distorsion et le nombre de vertex par chaîne. Cette méthode pourrait apporter une solution au codage à très bas débit. Evidemment, un compromis est nécessaire entre la qualité d'image tolérée et le débit de transmission; il serait aussi raisonnable de garder la mesure de distorsion quadratique proche de 1, sinon la pénalité allouée aux différentes segmentations devient excessive (les déformations des régions sont trop fortes).

Envisageons un très bas débit moyen, valant 16 kbits/s, pour la transmission de l'information audio-vidéo. La fréquence temporelle étant égale à 30 trames/s dans la plupart des applications envisagées, un sous-échantillonnage est effectué pour considérer seulement 10 trames/s au transmetteur. Donc, un maximum de 1600 bits par trame est disponible pour représenter toute l'information audio-vidéo. Si nous récapitulons les résultats obtenus sur la compression de cartes de segmentation, nous obtenons les points suivants: dans le cas des séquences simples de type "Miss America", il est possible de coder, sans ou avec perte, les segmentations intra et inter sous la barre des 1600 bits. En effet, le mode intra nécessite environ 1500 bits sans perte, 1000 avec perte, et le mode inter exige environ 800 bits sans perte et 400 bits avec perte. Un codage audio-vidéo à très bas débit serait donc réalisable, d'autant plus que le mode inter est plus fréquemment utilisé que le mode intra. Par contre, pour les séquences à plus grands mouvements (plus forte complexité) telles

que “Carphone” ou “Foreman”, les conclusions sont tout-autres; la compression est bien plus difficile puisque ces séquences possèdent beaucoup plus de points de contour (plus de régions). Là, il faut apporter des améliorations à la compression, à partir d’un codage avec perte probablement nécessaire car les méthodes sans perte ne compriment pas suffisamment.

## 6.1. Travaux futurs

Comme les résultats proposés ne sont pas encore suffisants, quelques suggestions au projet peuvent être apportées en vue d’augmenter la capacité de compression pour la représentation des cartes de segmentation.

Tout d’abord, signalons la remarque suivante: la performance de chaque technique étudiée a été spécialement basée sur le critère du nombre total de bits pour coder l’information de segmentation, étant donnée la volonté d’appliquer de telles méthodes au codage vidéo à très bas débit. Or généralement, la performance d’une méthode de codage peut être évaluée en fonction d’un modèle statistique pour les données. Ici, les cartes de segmentation ou les cartes d’état pourraient probablement être modélisées par des champs markoviens bidimensionnels. de telle sorte que la compression maximale serait dès lors évaluée sur ces modèles. Le développement de tels modèles serait donc souhaitable, malgré l’aspect encore “arbitraire” de l’obtention des cartes de segmentation (surtout intra-trame).

Ensuite, la procédure de segmentation intra-trame doit être paufinée; l’information de couleur, qui donne généralement plus de détails sur la scène que la luminance, devrait être incorporée à la détermination de la segmentation, pour offrir une procédure de segmentations jointes “basé-luminance” et “basé-couleur”. En fait, le groupe des Communications Visuelles de l’INRS y travaille, et de meilleurs

résultats sont à prévoir. De plus, cette approche rendrait cet algorithme bien plus fiable et robuste. Quant à la procédure de segmentation “basé-mouvement”, l’étape de fusion reste encore sous-optimale: plusieurs régions sont sur-segmentées. Tant que le suivi temporel ne sera pas inclus dans le traitement, on perdra de la précision au niveau du processus d’estimation du mouvement et de segmentation jointes, ainsi que la consistance entre les segmentations successives de la séquence.

Nous avons mentionné que l’on pourrait considérer, pour le codage sans perte CC, un processus de Markov d’ordre strictement supérieur à 1 pour modéliser les contours de région. Seulement, il faudrait s’assurer que la complexité de l’opération entreprise reste raisonnable. Signalons aussi que l’algorithme de codage avec perte peut être amélioré, en effectuant une étude plus poussée de certaines propriétés de l’image (allure des contours, complexité des chaînes...), afin d’appliquer une procédure d’approximation adaptative.

En outre, comme la tendance du projet est “basé-mouvement”, l’étape suivante consisterait à trouver la corrélation temporelle entre les contours approximatés des séquences d’images. Comme chaque contour approximaté est maintenant déterminé par ses vertex, nous pourrions estimer l’évolution temporelle “basé-mouvement” des vertex, en vue d’un suivi temporel des cartes de segmentation qui améliorerait certainement le taux de compression recherché. Ce point majeur n’a pas du tout été pris en considération, mais il semble indispensable si l’on souhaite optimiser, dans le système décrit à la Fig. 2.2, la procédure de compression des cartes de segmentation.

Finalement, en se référant aux études parallèlement menées au sein du groupe de recherche, la modification de la carte de segmentation dans le cas d’un codage avec perte (approximation polygonale optimale) cause une légère réduction de l’efficacité

---

du codage inter-image avec compensation du mouvement [47]. Aussi, le vrai critère de performance du codage des cartes de segmentation ne devrait pas être seulement axé sur le nombre total de bits alloués pour l'information de segmentation, mais devrait plutôt se focaliser sur la réalisation du système optimal de codage à très bas débit; ce système optimal cherche la meilleure répartition du débit alloué entre le mouvement, la segmentation et le résiduel (Fig. 2.2). Il serait alors souhaitable de considérer un critère de performance "basé-débit" ou "basé-taux" pour la compression des cartes de segmentation, en vue de la conception d'un système optimal de codage vidéo à très bas débit.



# Références

- [1] Didier Le Gall. A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, Avril 1991.
- [2] M. Liou. Overview of the px64 kbit/s Video Coding Standard. *Communications of the ACM*, Avril 1991.
- [3] L. Chiariglione. New Work Item Proposal for Very-Low Bitrate Audio-Visual Coding. *ISO/IEC JTC1/SC29/WG11*, Nov. 1992.
- [4] R. Schaphorst. Report of the ITU-TS WP 15/1 Special Rapporteur for Very-Low Bitrate Visual Telephony. *ISO/IEC JTC1/SC29/WG11*, Sept. 1993.
- [5] C. Reader et L. Chiariglione. MPEG-4 Proposal Package Description. *ISO/IEC JTC1/SC29/WG11*, Nov. 1994.
- [6] T. Wuyts, B. Deknuydt, S. Desmet, L. Van Eycken et A. Oosterlinck. Some Critical Remarks on 2-D Object-Based Coding. *COST 211ter European Workshop*, Déc. 1993.
- [7] S. Chae, J. Kim et R. Park. Video Coding by Segmenting Motion Vectors and Frame Differences. *Optical Engineering*, pages 870–876, Avril 1993.
- [8] C. Stiller et B. Huertgen. Combined Displacement Estimation and Segmentation in Image Sequences. *EUROPTO*, pages 23–56, Avril 1993.
- [9] C. Stiller. Object Oriented Video Coding Employing Dense Motion Fields. *ICASSP*, pages 273–276, 1994.
- [10] H. Sanson. Joint Estimation and Segmentation of Motion for Video Coding at Low Bitrates. *GRETSI*, Sept. 1993.
- [11] M. Kunt, A. Ikonopoulou et M. Kocher. Second Generation Image-Coding Techniques. *IEEE*, vol. 73:549–574, Avril 1985.

- 
- [12] M. Kunt, M. Bénéard et R. Leonardi. Recent Result in High-Compression Image Coding. *IEEE Trans. on Circ. and Syst.*, vol. 34:1306–1336, Nov. 1987.
- [13] H. Musmann, M. Hoetter et J. Ostermann. Object-Oriented Analysis-Synthesis Coding of Moving Images. *Signal Process: Image Comm*, vol. 1:117–138, Oct. 1989.
- [14] Y. G. Leclerc. Constructing Simple Stable Descriptions for Image Partitioning. *Int. Journal of Computer Vision*, vol. 3:73–102, 1989.
- [15] G. Labit et H. Nicolas. Compact Motion Representation based on Global Featured for Semantic Image Sequence Coding. *VCIP*, vol. 1605:697–708, 1991.
- [16] N. Diehl. Object-Oriented Motion Estimation and Segmentation in Image Sequences. *Signal Processing: Image Comm.*, vol. 3:23–56, 1991.
- [17] M. Gilge, T. Engelhardt et R. Mehlan. Coding of Arbitrarily Shaped Image Segments Based on a Generalized Orthogonal Transform. *Signal Processing: Image Comm.*, pages 153–180, 1989.
- [18] M. Goetze et H-J. Pfeifer. Image Coding by Direction Adapted Transforms. *Signal Processing*, 1983.
- [19] M. Eden et M. Unser. Polynomial Representation of Pictures. *Signal Processing*, vol. 10:385–393, 1986.
- [20] L. Perrin. Codage d'un segment d'image de forme arbitraire par transformée orthogonale adaptée. Rapport Technique 94-33, INRS-Télécommunications, Verdun, Québec, 1994.
- [21] U. Franke et R. Mester. Region-Based Image Representation with Variable Reconstruction Quality. *SPIE*, vol. 1001, Nov. 1988.
- [22] K. McConnell, D. Bodson et R. Schaphorst. *Digital Facsimile Technology & Applications*. Artech House, 1992.
- [23] R. Hummel et S. Zucker. On the Foundations of Relaxation Labelling Processes. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5:267–287, 1983.
- [24] M. Ferguson. Notes on Optimal Source Coding. Rapport Technique 85-39, INRS-Télécommunications, Verdun, Québec, 1985.
- [25] S. Golomb. Run-Length Encodings. *IEEE Trans. Info. Theory*, pages 399–401, Juil. 1966.
- [26] W. Pennebaker et J. Mitchell. *JPEG: Still Image Data Compression Standard*. Van Nostrand Reinhold, 1990.



- 
- [27] R. B. Arps et T. K. Truong. A multi-purpose VLSI Chip for Adaptive Data Compression of Bi-level Images. *IBM J. Res. Develop.*, Nov. 1988.
- [28] G. Langdon et J. Rissanen. Compression of Black-White Images with Arithmetic Coding. *IEEE Trans. on Comm.*, pages 858–867, Juin 1981.
- [29] G. Langdon. An Introduction to Arithmetic Coding. *IBM J. Res. Develop.*, pages 135–149, Mars 1984.
- [30] F. Ono et S. Kino. Bi-level Image Coding with Melcode-Comparison of Block Type Code and Arithmetic Type Code. *GLOBECOM*, Nov. 1989.
- [31] W. Pennebaker et J. Mitchell. Probability Estimation for the Q-Coder. *IBM J. Res. Develop.*, pages 737–759, Nov. 1988.
- [32] J. Cleary, I. Witten et R. Neal. Arithmetic Coding for Data Compression. *Commun. of the ACM*, pages 520–540, Juin 1987.
- [33] M. Eden et M. Kocher. On the Performance of a Contour Coding Algorithm in the Context of Image Coding. *Signal Processing*, 8:381–386, Mai 1985.
- [34] M. Hoetter. Object-Oriented Analysis-Synthesis Coding Based on Moving 2-D Objects. *Signal Processing*, vol. 2:409–428, 1990.
- [35] C. Choo et H. Freeman. Comparative Precision Analysis of Chain Coding Families for Line Drawing Encoding. *SPIE*, vol. 1251:194–205, 1990.
- [36] S. D. Kim et J. H. Lee. A New Chain Coding Algorithm for Binary Images using Run-Length Codes. *Computer Vision, Graphics and Image Proc.*, vol. 41-1:114–128, Jan. 1988.
- [37] K. Liu et R. Prasad. Performance Analysis of Differential Chain Coding. *European Trans. on Telecom*, vol. 3-4:323–330, Juil. 1992.
- [38] N. Chang et S. Kulkarni. Adaptive Chain Coding for arbitrary Curves. *SPIE*, vol. 1830:296–307, 1992.
- [39] A. Johannessen et R. Prasad. Coding Efficiency of Multiring Differential Chain Coding. *IEE Proceedings-1*, vol. 139-2:224–232, Avril 1992.
- [40] T. Cover et J. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.
- [41] T. Pavlidis. *Algorithms for Graphics and Image Processing*. Computer Science Press, 1982.

- [42] S. Ali et R. Burge. A New Algorithm for Extracting the Interior of Bounded Regions based on Chain Coding. *Computer Vision, Graphics and Image Proc.*, vol. 43-2, Août 1988.
- [43] Z. Cai. Restoration of Binary Images using Contour Direction Chain Codes Description. *Computer Vision, Graphics and Image Proc.*, vol. 41-1, Jan. 1988.
- [44] J. Shim, S. Cho et Y. Ha. A New Interior-Filling Algorithm Based on Binary Boundary Coding. *Journal of Korean Inst. of Telematics and Elect.*, vol. 26-11:238–241, Nov. 1989.
- [45] J. Perez et E. Vidal. Optimum Polygonal Approximation of Digitized Curves. *Pattern Recognition Letters*, vol. 15:743–750, Août 1994.
- [46] C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, N.Y., 1978.
- [47] V.-N. Dang, A. R. Mansouri et J. Konrad. Motion Estimation for Region-Based Video Coding. *ICIP*, Oct. 1995.