

Optimisation débit-distorsion d'un encodeur vidéo par une représentation flexible du mouvement

Daniel Lauzon

5 février 2001

Table des matières

1	Introduction	1
1.1	Motivation	1
2	État de la recherche actuelle	2
2.1	Fondements et problématique	2
2.1.1	Représentation de séquences d'images	3
2.1.2	Structure générale d'un codec vidéo	4
2.1.3	Débit et distorsion	5
2.2	Analyse débit-distorsion	6
2.2.1	Répartition du débit	9
2.2.2	Estimation du mouvement	10
2.2.3	Quantification des vecteurs de mouvement	11
2.2.4	Structure hiérarchique de blocs	12
2.2.5	Régions adaptatives	13
2.2.6	Sélection du mode de codage	14
2.2.7	Programmation Dynamique	16
2.2.8	Prédiction multi-trame	18
2.2.9	Analyse stochastique de l'efficacité de la compensation en mouve- ment	19
2.3	Estimation et compensation en mouvement	20

3	Optimisation débit-distorsion	23
3.1	Méthode des multiplicateurs de Lagrange	24
3.2	Choix de λ	25
3.3	Problème simplifié : codage d'une image	26
3.3.1	Facteur de quantification fixe	27
3.3.2	Facteur de quantification variable	28
3.3.3	Modélisation par moindres carrés	28
3.3.4	Modélisation par l'enveloppe convexe	29
3.4	La pratique	31
4	Codage de cartes étiquettes	32
4.1	Modélisation de la source	32
4.2	Modèle sans mémoire	33
4.3	Modèle de premier ordre	36
4.3.1	Probabilités de transition complètes	36
4.3.2	Modèle à nombre de paramètres réduit	38
4.4	Modèle de deuxième ordre	42
4.4.1	Probabilités de transition complètes	43
4.4.2	Modèle à nombre de paramètres réduit	44
4.5	Généralisation à des voisinages élargis	45
4.5.1	Probabilités de transition complètes	46
4.5.2	Modèle à nombre de paramètres réduits	47
4.5.3	Troncation des contextes	49
4.5.4	Probabilités marginales	49
4.5.5	Exemple	50
5	Représentation du mouvement	53
5.1	Composantes de la représentation du mouvement	53
5.1.1	Les régions	53

5.1.2	Le mouvement	55
5.1.3	L'image ou le résiduel	57
5.2	Quantification en amplitude du mouvement	58
5.2.1	Solution du problème	59
5.3	Représentation du mouvement par un dictionnaire de modèles	61
5.3.1	Représentation	63
5.3.2	Estimation	64
5.3.3	Affectation d'étiquettes	64
5.3.4	Ré-estimation des modèles	65
5.4	Codage de la carte d'étiquettes	66
5.5	Codage entropique conditionnel contraint sur le taux	67
6	Simulations et analyse	68
6.1	Quantification en amplitude des vecteurs de mouvement	68
6.2	Codage de cartes d'étiquettes	68
6.3	Représentation du mouvement à résolution variable	74
6.4	Représentation du mouvement par un dictionnaire de modèles	74
6.5	Intégration dans un codec optimal débit-distorsion	79
6.5.1	Compensation à partir de prédicteurs de haute qualité	79
6.5.2	Simulation a distorsion constante	80
7	Conclusions	82
7.1	Contributions	82
7.2	Travaux futurs	82

Chapitre 1

Introduction

Incomplet

1.1 Motivation

Le problème est ouvert. Pourquoi faut-il l'examiner.

La représentation de séquences d'images pour des fins de transmission ou de stockage pose une foule de défis. C'est un vaste domaine de recherche qui suscite énormément d'intérêt.

Chapitre 2

État de la recherche actuelle

Dans la plupart des standards de compression vidéo (MPEG, H.263, etc.), souvent nommés codeurs hybrides, il s'agit de codage vidéo par transformée d'un résiduel compensé en mouvement. Beaucoup de travaux ont été effectués pour analyser et raffiner les diverses composantes du codage vidéo. Par exemple, plusieurs auteurs ont contribué à l'élaboration de nouvelles représentations du mouvement pour donner une flexibilité accrue à leur modèle et donc un plus haut degré de liberté pour ensuite passer à l'optimisation du modèle. Pour préciser ces notions d'optimalité de la performance d'un codec, on fait généralement appel à des notions de débit-distorsion. Plusieurs auteurs ont considéré des critères débit-distorsion dans différents contextes que nous tenterons de présenter succinctement pour mettre nos méthodes en relation avec les travaux courants de la communauté.

2.1 Fondements et problématique

Nous souhaitons utiliser une approche basée sur les notions de débit-distorsion pour adresser deux points fondamentaux :

- Comment effectuer une répartition optimale du débit entre les composantes de représentation : *e.g.* la répartition du débit qui sert au codage de l'information de mouvement et celui qui sert au codage du résiduel de prédiction de l'image.

- Comment choisir les paramètres d’opération d’un codec qui soient optimaux au sens débit-distorsion : *e.g.* choisir la taille des blocs utilisés pour la compensation en mouvement ou la précision des vecteurs de mouvement.

Dans les deux cas, nous désirons trouver une solution au problème général

$$\min_{\mathbf{p}} D(\mathbf{p}) \quad \text{s.c.} \quad R(\mathbf{p}) = R_0 \quad (2.1)$$

où $\mathbf{p} = (p_i)$ est un vecteur de paramètres (variables indépendantes) et les fonctions $R(\mathbf{p})$ et $D(\mathbf{p})$ représentent respectivement le débit et la distorsion associés aux paramètres \mathbf{p} .

Effectuons donc un survol des travaux courants qui adressent des problèmes semblables.

2.1.1 Représentation de séquences d’images

Les systèmes décrits dans cet exposé traitent de séquences d’images. Dans sa forme la plus simple une séquence d’images associe une valeur d’intensité lumineuse (possiblement de couleur) avec un ensemble de positions spatio-temporelles. Pour le traitement numérique on se limite à un ensemble fini (ou comptable) de telles positions, et ces positions sont normalement décrites par un réseau d’échantillonnage.

La représentation *naturelle* d’une séquence d’image est donc une simple énumération de ces valeurs d’intensité pour chacune des positions du réseau.

Si par contre on veut obtenir la représentation la plus compacte possible, nous nous rendons compte que la représentation naturelle n’est pas particulièrement efficace parce que cette dernière renferme une très grande redondance spatio-temporelle. Il est possible de définir des représentations alternatives qui présentent des caractéristiques différentes. Le but d’un encodeur/décodeur est de passer d’une représentation à une autre tel que montré dans la figure 2.1.

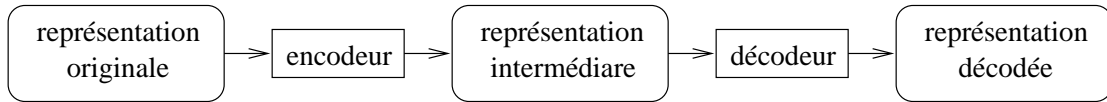


FIG. 2.1 – Représentation des séquences

2.1.2 Structure générale d'un codec vidéo

Dans l'analyse d'un codec il est souvent utile de distinguer différentes composantes de la représentation encodée. La figure 2.2 illustre la relation entre ces composantes pour un codec typique. On y voit principalement la boucle de prédiction/compensation en mouvement, le codage par transformée (T) et le codage entropique.

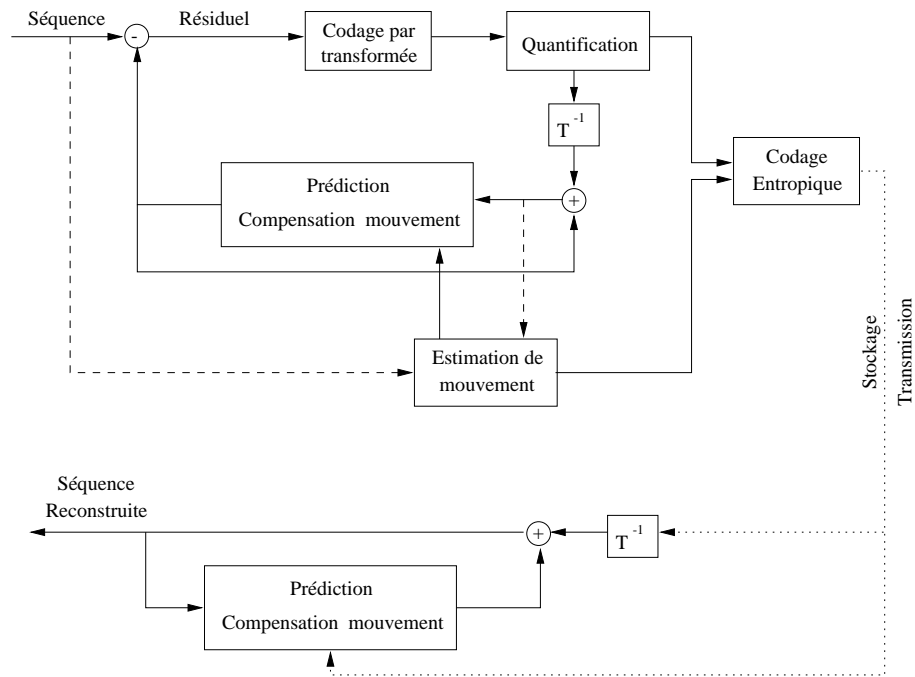


FIG. 2.2 – Structure du codec

On réfère souvent aux encodeurs qui utilisent une boucle de prédiction (compensation en mouvement) suivi d'un codage par transformée du résiduel de prédiction comme codeurs *hybrides*. Parmi les encodeurs qui adoptent cette approche nous avons quand même une grande flexibilité pour implanter les différentes composantes.

Représentation de l'image ou du résiduel. On associe normalement des valeurs d'intensité à chaque point d'un réseau d'échantillonnage. On fait souvent appel à un codage par transformée (comme la DCT) pour encoder ces données.

Représentation du mouvement Le mouvement sert à relier des positions dans des images différentes. On se sert typiquement de cette information pour former un prédicteur d'une image à encoder à partir d'une autre image précédemment encodée.

Représentation des régions Les régions définies par un modèle peuvent servir simultanément à plusieurs buts *e.g.* le support d'un codage par transformée pour la composante image ou encore le support pour la compensation en mouvement.

2.1.3 Débit et distorsion

On associe à la représentation encodée un débit qui indique les ressources nécessaires à sa représentation (*e.g.* bits/s. pour un canal ou capacité totale pour le stockage).

Comme la représentation décodée n'est pas nécessairement identique à l'originale, on caractérise cette dégradation par une mesure qu'on nomme distorsion.

Il existe une relation étroite entre le débit et la distorsion. Cette relation est centrale au codage de séquences d'images, où nous voulons typiquement minimiser la distorsion sujet à une contrainte sur le débit (ou de manière équivalente de minimiser le débit sujet à une contrainte sur la distorsion).

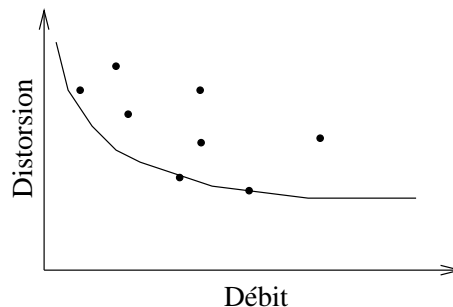


FIG. 2.3 – Courbe débit-distorsion

Pour chaque représentation admissible dans un codec, on peut définir un point composé

du débit et de la distorsion associée. L'enveloppe de tous ces points constitue la courbe limite débit-distorsion (*optimal rate-distortion curve*). Cette courbe est bien étudiée dans plusieurs contextes simplifiés, par exemple dans l'ouvrage de référence [GG91].

2.2 Analyse débit-distorsion

Depuis près de cinquante ans maintenant, en commençant par l'oeuvre de Shannon [Sha48], nous essayons de formaliser les notions essentielles de la communication. Bien que les systèmes de communication aient évolué considérablement depuis ce temps, l'objectif de transmettre une information maximalelement *dense* tout en étant *fidèle* est toujours centrale. Pour un aperçu général des méthodes débit-distorsion relatives au codage vidéo voir les articles [SW98, OR98].

L'objet de l'analyse débit-distorsion est donc d'éliminer le plus possible la redondance d'une source d'information, ou d'utiliser le moins de bits possible à sa représentation pour une qualité de reproduction désirée. Même dans le contexte spécifique du codage vidéo il faudra préciser ce qu'on entend par source, et par qualité. Une source pourrait être une réalisation bien précise d'un signal vidéo, d'un format précis, ou elle pourrait être un ensemble de sources possédant des caractéristiques semblables. Et dans ce deuxième cas à quelles caractéristiques allons nous nous attarder. Pour rendre la modélisation tractable on aura souvent recours à des modèles simplifiés.

De la même façon comment définir rigoureusement la notion de qualité ? La réponse à ces questions est centrale à la formulation du problème d'optimisation. Elle guidera notre choix d'algorithmes et d'outils, et doit nécessairement toujours être à l'esprit lorsque nous obtenons des résultats analytiques.

Pour ce qui est de la distorsion, on distingue d'abord les cas de codage avec et sans perte. Un codage sans perte reproduit la source exactement, tandis que le codage avec perte peut tolérer un signal reproduit différent de la source. Pour le codage avec perte nous voulons donc obtenir une mesure de la distorsion, qui est souvent reliée à l'utilisation que nous

voulons faire du signal reproduit. Il est concevable par exemple, qu'un codeur destiné à des images médicales ait non seulement un plus grand besoin de fidélité, mais que la mesure de la distorsion soit différente. Le plus souvent, nous désirons lier la mesure de distorsion à la qualité subjective perçue par un observateur humain. Le système visuel humain est cependant très complexe, dans ses mécanismes de perception de la couleur, du mouvement relatif, et généralement du *réalisme* perçue. Beaucoup de travaux dont ceux du groupe de travail VQEG de l'ITU visent à comprendre la nature de la perception de la qualité, et de produire de mécanismes dits objectifs pour sa mesure. Cependant le problème est toujours ouvert et encore une fois on a plutôt recours à des simplifications qui rendent les modèles débit-distorsion mathématiquement tractables. Parfois il est possible d'inclure des propriétés du système visuel humain, comme la pondération perceptuelle des différentes bandes de fréquences dans le codage d'images. Bien qu'il faille toujours garder ces qualifications en tête, il se trouve que des mesures très simples comme l'erreur quadratique moyenne (MSE), donnent des résultats très utiles que même des modèles perceptuels complexes ont difficulté à battre. Il faut surtout faire attention en comparant des systèmes qui produisent des dégradations de nature fort différentes. Par exemple il est très difficile de comparer une mesure de rapport signal à bruit d'un système de transmission analogique conventionnel qui produit une dégradation graduelle (neige) à celui qui utilise des techniques de codage vidéo tels que MPEG et qui produit des dégradations hautement structurées comme des effets de blocs, ou des sections de trames manquantes.

La première grande utilité de l'analyse débit-distorsion sera d'analyser les bornes de performance d'un système pour une classe de sources. Sauf pour des modèles très simples de sources statistiques, il n'y a pas de résultat universel qui décrit la courbe débit-distorsion optimale. Cependant le problème des quantificateurs scalaire (et vectoriels) optimaux, où les sources sont représentés par une séquence de symboles encodés indépendamment (ou en blocs) sont bien étudiés et compris (*e.g.* [SG88, CLG89a, CL90, GG91, dGP95, WWW99]). Ces résultats généraux reposent souvent sur le comportement asymptotique du codage quand la longueur des blocs de symboles encodés devient très grand ou supposent que

le nombre de niveaux de quantification est très grand. Même dans ces contextes simplifiés où les résultats nous donnent une borne sur la courbe, il n’y a pas toujours de méthode pour construire un codeur pratique qui atteigne la borne. On discute dans [OR98] de l’importance de considérer simultanément les deux questions suivantes :

- Pour la source qui nous intéresse, quel est le modèle probabiliste ou autre qui définit cette source ?
- Pour ce modèle de la source, et étant donné les bornes que nous indique l’analyse, pouvons nous construire un codeur pratique qui s’approche de ces bornes ?

On montre ensuite avec un exemple frappant deux approches au codage d’images fixes. La première basé sur un modèle statistique Gaussien de la décomposition en sous-bandes d’une image, montre une certaine borne inférieure atteignable asymptotiquement avec un codage basé sur une longueur de blocs source infini. On montre ensuite que le codeur réalisable de type ondelettes-EZW [Sha93, SP96b] supplante la borne inférieure du premier codeur d’une marge considérable. Ce qui démontre bien qu’il n’est pas suffisant de montrer l’optimalité du système de codage, il faut bien choisir le modèle de la source, ou tout au moins comprendre que si ce modèle n’exploite pas toutes les propriétés de la sources, l’optimalité du codeur résultant au sens plus large sera compromise.

Le deuxième usage de l’analyse débit-distorsion est la caractérisation de la performance réalisée par un ou plusieurs systèmes. On distingue souvent l’ensemble des points débit-distorsion réalisables des bornes analytiques en les nommant *courbes débit-distorsion opérationnelles*. Les courbes opérationnelles sont donc un bonne façon de comparer plusieurs systèmes de codage, ou encore de caractériser un codeur sur un ensemble de points d’opération.

L’analyse débit-distorsion sera donc utile dans le contexte plus restreint où le système de codage a été déterminé, et nous désirons faire la sélection de paramètres d’opération du codeur pour une source ou une classe de sources. La formulation la plus répandue pour le problème d’optimisation débit-distorsion est un forme de relaxation lagrangienne telle qu’utilisée dans [CLG89b, CLG89a], et que nous formalisons dans le chapitre 3. Dans une

incarnation simple, comme la sélection des paramètres de quantification des blocs d'une image, comme c'est le cas pour le codage JPEG, ou le codage d'une image INTRA de MPEG, on considère que l'expression débit-distorsion à minimiser est composée de sous-problèmes indépendants (*les blocs*). Lorsque il y a une dépendance entre les unités de codage (*blocs, images*), le problème devient beaucoup plus complexe. Une autre formulation du problème, qui tente d'adresser l'effet de la dépendance, repose sur la *programmation dynamique*.

Voyons maintenant un échantillon de travaux qui ont utilisé ces techniques dans le contexte du codage vidéo.

2.2.1 Répartition du débit

Rate Constrained Motion Estimation [Gir94]

Dans cet article l'auteur définit de manière très générale les propriétés nécessaires à l'optimalité au sens débit-distorsion des différentes composantes de l'information encodée.

Pour un débit total $R = R_1 + R_2 + \dots + R_N$ composé de différentes composantes indépendantes, on y dérive la condition nécessaire d'optimalité

$$\frac{\partial D}{\partial R_i} = \frac{\partial D}{\partial R_j} \quad \forall i, j \in 1, \dots, N \quad (2.2)$$

L'auteur discute ensuite du sens de cette condition dans le contexte de la précision (quantification en amplitude) des vecteurs de mouvement. De manière générale conclue-t-il, un codec *donné* utilisera une représentation moins précise du mouvement à plus faible débit.

Il faut bien comprendre la portée limitée de cet énoncé, qui décrit la relation entre les débits alloués au mouvement, au résiduel de prédiction, et aux contours pour un codec. Cette analyse ne prédit pas de façon générale que tous les codecs devraient utiliser moins de bits que MPEG (un codec spécifique) pour la représentation du mouvement lorsqu'il sont opérés à faible débit.

De plus dans cet article on décrit des simulations qui appliquent ce principe pour un codec basé sur des régions. Les contours sont encodés avec des courbes *B-spline*.

On utilise le critère d'optimisation pour arrêter le processus de subdivision progressive de l'image en sous régions. On rapporte un gain maximal de 3dB pour une séquence test encodée à 15 kbps par rapport à un codec semblable qui utilise des blocs de dimension fixe.

2.2.2 Estimation du mouvement

An Efficient Motion Estimation Technique Based on a Rate-Distortion Criterion [CKS95, CKS96]

Dans ces travaux, les auteurs examinent le critère de sélection du vecteur de mouvement associé à un bloc, dans les approches classique d'estimation de mouvement basé sur des blocs (BMA). Ils remarquent que ce critère de sélection tente de minimiser le résiduel de prédiction, mais ne tient pas compte de la quantification et du codage de ce résiduel. Ils modifient donc le critère de minimisation utilisé dans l'estimation de mouvement pour le remplacer par une relaxation lagrangienne qui tient compte à la fois du résiduel de prédiction, de sa quantification, ainsi que du coût de codage du résiduel et des vecteurs de mouvement. Notant que le codage inter-trame est parfois moins efficace que le codage intra-trame pour certains blocs,, on permet en plus la sélection du mode de codage inter-intra. L'intégration de cette méthode d'estimation de mouvement à un codec H-261 (PVRG de Stanford) mène à une réduction du débit de l'ordre de 25% pour une distorsion équivalente, ou encore d'une réduction du rapport signal à bruit de l'ordre de 2-3 dB pour un taux équivalent. Les expériences sont menés sur un petit ensemble de séquences standard en format QCIF (Miss America,Suzie).

On ne tiens pas compte de la dépendance inter-bloc du codage des vecteurs de mouvement. La sélection du mode de codage est prise en considération en permettant deux modes possibles, codage INTER, et codage INTRA. Essentiellement ces travaux remplacent strictement l'algorithme d'estimation du mouvement et le codec utilisé n'est pas autrement modifié.

Voyons maintenant une approche semblable où l'on considère la précision des vecteurs de mouvement estimés.

2.2.3 Quantification des vecteurs de mouvement

Dans [Gir93], l'auteur analyse l'efficacité de la compensation en mouvement à des précisions sous-pixel. De fait la compensation en mouvement par déplacement fractionnaire est posé dans le cadre plus général d'un filtrage de l'image prédictive que nous reverrons dans la section 2.2.9 de ce chapitre. On analyse aussi l'effet de la précision des vecteurs de mouvement sur la qualité de la prédiction. La précision optimale expérimentée, dans le codage des signaux de télévision typiques est de $\frac{1}{4}$ de pixel tandis qu'elle est de $\frac{1}{2}$ pixel pour des source de type vidéophone. Voyons comment cela est adressé dans le cadre d'une optimisation débit-distorsion.

On the Optimal Motion Vector Accuracy for Block-Based Motion-Compensated Video [RCN96a, RCN96b]

Dans ces articles, les auteurs décrivent une méthode pour déterminer la précision optimale des vecteurs de mouvement. Cette précision (quantification en amplitude des vecteurs à $1, \frac{1}{2}, \dots, \frac{1}{16}$ pixels) peut être globale (une précision pour tous les blocs) ou adaptative (chaque bloc a sa propre précision).

D'après le modèle décrit dans l'article, les auteurs dérivent les valeurs optimales pour la précision des vecteurs de mouvement de façon analytique (en tenant compte de certaines simplifications). La précision des vecteurs de mouvement est représentée par un pas de quantification $\Delta_{x,i}$ ($\Delta_{y,i}$) pour la quantification en amplitude de la composante horizontale (verticale) du vecteur associé au bloc i .

- La distorsion est essentiellement contrôlée par un paramètre Q de quantification globale (quantification scalaire uniforme du résiduel de prédiction sans codage par transformée comme la DCT)
- On suppose que les résiduels ont une distribution Laplacienne dont la variance est estimée par l'énergie du résiduel de prédiction pour un bloc i (E_i).

- On modélise cette énergie de résiduel en fonction de la précision du vecteur de mouvement utilisé :

$$E_i \approx A_i \Delta_{x,i}^2 + B_i \Delta_{y,i}^2 + C_i \quad (2.3)$$

- Les coefficients $\{A_i, B_i, C_i\}$ sont estimés par moindres carrés pour chaque bloc.

Cette approche permet de minimiser l’expression du débit total $R(Q, \Delta_{x,i}, \Delta_{y,i})$ en fonction des pas de quantification $\Delta_{x,i}$ et $\Delta_{y,i}$.

Les paramètres à optimiser (\mathbf{p}) par rapport à notre formulation initiale sont : $\Delta_{x,i}$, $\Delta_{y,i}$ et Q .

Les résultats démontrent que la précision optimale donne un gain appréciable par rapport à l’estimation à précision 1 ou 1/2 pixel qui est typiquement utilisée, mais que l’adaptation de la précision optimale pour chaque bloc ne donne aucun gain appréciable.

2.2.4 Structure hiérarchique de blocs

Optimal Quadtree for Variable Block Size Motion Estimation [Lee95, Lee98]

Dans cet article, l’auteur décrit une méthode pour déterminer simultanément une structure de blocs hiérarchique, les vecteurs de mouvement et les facteurs de quantification associés à ces blocs.

Ici le problème est directement posé comme une minimisation globale de la distorsion sujet à une contrainte sur le débit. On utilise une relaxation lagrangienne pour transformer le problème en minimisation sans contrainte.

L’algorithme présenté utilise un élagage ordonné qui suppose la capacité de calculer le coût associé aux sous-arbres d’un noeud de manière indépendante. Un noeud représente la division/non-division d’un bloc en sous-blocs. Pour chaque noeud on peut choisir un facteur de quantification Q optimal et la décision de séparer ou d’unifier un noeud est prise en comparant le coût d’un noeud parent à celui de ses enfants.

Rate-constrained hierarchical motion estimation using BFOS tree pruning [BC96]

Dans l’article [BC96] on propose une méthode très semblable pour l’estimation du

mouvement sur un structure de blocs hiérarchique mais on ne fait pas intervenir le coût de codage du résiduel de prédiction.

2.2.5 Régions adaptatives

Avec la venue du codage vidéo à base d'objets, il devient possible, et même nécessaire de considérer des formes plus générales de régions, comme support à la compensation en mouvement et au codage par transformée. Encore une fois l'efficacité de ces représentations, ainsi que l'optimisation de leur paramètres peut être posé dans le cadre de l'optimisation débit distorsion.

Rate-constrained contour representation for region based motion compensation [SSG96]

Dans cet article on propose d'encoder une séquence par une compensation en mouvement basé sur un nombre variable de régions et l'encodage du résiduel de prédiction par transformée DCT et par codage entropique. Le contour de chacune des régions est représenté par des *splines* et défini par un nombre variable de points de contrôle.

L'algorithme permet d'ajouter successivement des régions pour compensation en mouvement. Lorsqu'une région est ajoutée, on optimise la précision de son contour en faisant varier la position et le nombre de points de contrôle qui définissent cette nouvelle région. Une fois la région ajoutée, sa représentation est fixée définitivement (algorithme de type *greedy*).

Ici encore on utilise une formulation lagrangienne pour représenter le problème. Lorsqu'une région candidate (ρ) est ajoutée à la représentation elle donne lieu à une nouvelle valeur de distorsion pour la représentation et aussi à un nouveau débit.

$$\Delta D_\rho = D_{\text{sans } \rho} - D_{\text{avec } \rho} \quad (2.4)$$

$$\Delta R_\rho = R_{\text{sans } \rho} - R_{\text{avec } \rho}$$

La décision d'ajouter la région ρ à la représentation courante est basée sur le rapport $\frac{\Delta D_\rho}{\Delta R_\rho}$ entre les variations dans la distorsion et dans le débit en ajoutant la région ρ au modèle. Pour que la région soit incluse dans la représentation, ce rapport doit excéder le multiplia-

teur de Lagrange associé à l'itération courante de l'algorithme. Le multiplicateur indique donc un seuil minimal d'efficacité pour l'ajout d'une région.

Dans *Operationally Optimal Vertex-Based Shape Coding*[SMK98] on présente aussi une représentation dite optimale, au sens débit-distorsion, des contours d'objets donnés (la segmentation n'est pas considérée) pour le codage INTRA dans un codeur MPEG-4.

2.2.6 Sélection du mode de codage

Dans la plupart des codeurs vidéo standards, il y a un support syntaxique pour ce qu'on appelle des modes d'opération. Ces modes spécifient pour une trame ou un bloc, le type de codage prédictif qui sera utilisé. Ces modes comprennent des variantes de prédiction simple et bi-directionnelle, telles, les modes INTRA (sans prédiction), NOMC (compensation en mouvement avec déplacement nul), mode INTER (prédiction avant, arrière, bi-directionnelle), codage par transformée basé trame/champ, parmi d'autres. Dans l'opération typique de ces codeurs, le contrôle du débit et la sélection du mode de codage sont fait indépendamment. Une approche plus générale qui prend en considération le point d'opération exacte du codeur, considérant le mode de prédiction conjointement avec la quantification et le codage du résiduel offre donc une possibilité d'amélioration.

C'est une telle méthode que l'on décrit dans *MPEG Coding Performance Improvement by Jointly Optimizing Code Mode Decisions and Rate Control* [SKJ97]. Un des problèmes qui y est relevé est la dépendance du codage de blocs successifs sur les décision du mode de codage des blocs précédents. Décrivant d'abord une formulation optimale qui requiert l'examen M^S combinaisons où on a M modes possibles dans S blocs à coder, et considérant que cette approche non-utilisable en pratique, on propose une méthode approximative de sélection successive (*greedy*) qui examine plutôt MS options. On présente aussi une méthode encore plus simple où le coût de codage du résiduel est modélisé en fonction de l'énergie du résiduel de prédiction pour chaque mode de prédiction. Les résultats des simulations suggèrent une réduction de 10 à 15% par rapport au *Test Model MPEG-2* pour le matériel test MPEG-2 codé à environ 5 Mb/s.

Dans l'opération d'un codec de type MPEG-2 en plus du mode de codage du bloc, il y a aussi un mode de codage d'une trame, communément appelés trames I,P et B. Les trames I (intra) sont codées sans référence d'autres trames, les trames P (prediction avant seulement) sont codées en référence à la trame I ou P précédente, et finalement les trames de type B (bi-directionnelles) sont codées en référence aux 2 trames I ou P les plus proches tel qu'illustré la figure 2.4.

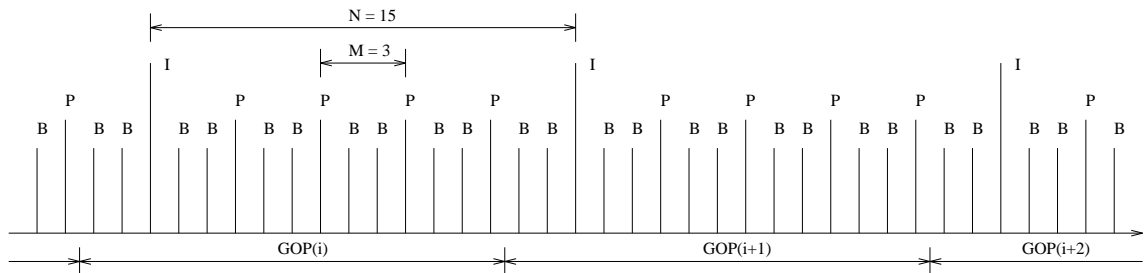


FIG. 2.4 – Trames I,P,B de MPEG-2

L'opération typique d'un codeur MPEG-2 fixe une structure régulière de groupe de trames (GOP) qui est caractérisé par la distance N entre deux trames I, et par le nombre M de trames B entre deux trames I ou P. Cependant la syntaxe MPEG-2 permet d'organiser les types de trames de manière très flexible et irrégulière. L'article suivant, *Rate-Distortion Optimized Frame Type Selection for MPEG Encoding* [LD97], tente de déterminer une structure optimale des types de trames par une formulation débit-distorsion. Comme dans l'article précédent la complexité survient de la nature dépendante des paramètres à être déterminés. En observant que le coût lagrangien de codage d'une trame est généralement fonction monotonique du pas de quantification de la trame référence, ils arrivent à simplifier le nombre de combinaisons de structure de trames à considérer. La détermination de la structure du GOP est représentée comme un problème d'optimisation débit-distorsion qu'on solutionne par une relaxation lagrangienne. Le résultat de simulations indiquent un gain marginal pour la séquence Tennis, et aucon gain appréciable pour la séquence Football.

La détermination des paramètre optimaux en présence de dépendance entre ces choix

mène souvent une explosion dans la complexité du problème à résoudre. Dans ce cas on passe parfois par dessus cette complexité en proposant une solution sous-optimale qui ignore cette dépendance. Mais il est aussi possible dans certains cas de tenir compte d'une dépendance particulièrement structurée et conserver une approche optimale.

2.2.7 Programmation Dynamique

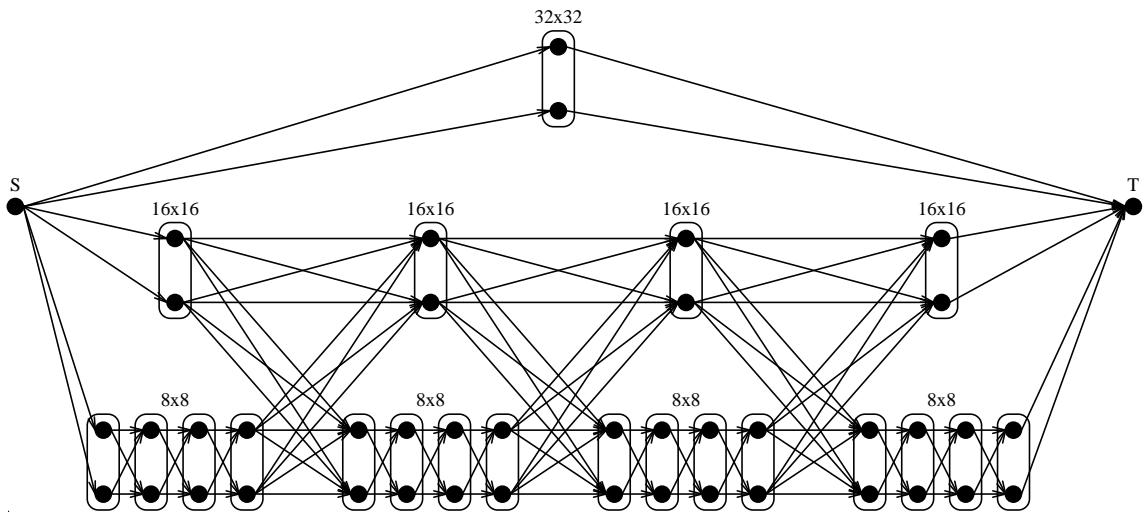


FIG. 2.5 – Treillis à niveaux multiples

Dans la structure de dépendance entre les paramètres d'opération on distingue deux types de cas. Illustrons ceci par deux exemples. Dans le premier cas considérons la nature de la dépendance du choix d'un vecteur de mouvement qui dépend d'un prédicteur dans une boucle de prédiction récursive. Clairement le choix optimal d'un vecteur de mouvement dépend de la trame reconstruite du prédicteur, qui à son tour dépend du choix de plusieurs vecteurs de mouvement, qui dépendent à leur tour d'une trame reconstruite, et ainsi de suite. Dans ce cas nous observons que le nombre de combinaisons des vecteurs de mouvement et de facteurs de quantification du résiduel augmente exponentiellement avec le nombre de niveaux de dépendance. En contrepartie considérons le choix du pas de quantification d'un macro-bloc dans MPEG-2 qui dépend du choix du quantificateur du bloc précédent seulement. Dans ce cas le graphe de dépendance forme un treillis dont la taille est constante

à chaque niveau de dépendance. Dans ce cas on peut représenter la combinaison des paramètres comme un chemin à travers un graphe où les noeuds représentent des choix de paramètre à un niveau et le poids associé à un arc est le coût lagrangien de codage associé à la combinaison des deux noeuds qu'il joint.

Ainsi le choix de la combinaison optimale des paramètres revient à trouver le chemin le plus court du graphe pour lequel il existe des algorithmes efficaces (basés sur le principe d'optimalité de Bellman). La solution de problèmes qui ont une telle formulation est souvent nommé *programmation dynamique*.

Dans *An Optimal Quadtree-Based Motion Estimation and Motion-Compensation Interpolation Scheme for Video Compression* [SK97, SK98], l'auteur raffine cette idée. Ici on essaie d'estimer simultanément une structure hiérarchique de blocs et un vecteur de mouvement associé à chacun de ces blocs en tenant compte de la dépendance du coût de codage des vecteurs qui sont codés différenciellement. On voit donc dans la figure 2.5 la représentation de la structure hiérarchique des blocs ; les parcours passent soit par un bloc parent, soit par ses quatre enfants. On voit aussi le treillis de dépendance entre les choix de vecteurs de déplacement successifs représentés par les arcs. L'auteur réussit donc à résoudre le problème de façon optimale au sens débit-distorsion. Des simulations ont été réalisées pour comparer la performance du système proposé relativement au modèle TMN4 de H.263 pour des séquences test typiques QCIF (Mother and Daughter, Foreman, Miss America) et l'on rapporte des gains de jusqu'à 25% sur le débit pour une distorsion équivalente. On ne considère pas dans l'article le codage du résiduel associé à la prédiction, l'article vise seulement l'estimation de la structure hiérarchique des blocs et l'estimation de vecteurs associés.

Dans *Rate-Distortion Optimal Motion Estimation Algorithm for Motion-Compensated Transform Video Coding* [CW96, CW98], on considère une structure de blocs régulière et fixe mais on considère en plus du choix des vecteurs de mouvement, la détermination du facteur de quantification du résiduel de prédiction. Comme le codage des vecteurs est effectué de manière différentielle au vecteur du bloc précédent le problème de dépendance refait

surface. Un compromis entre un algorithme *greedy* et la solution qui tient compte d'une solution optimale basée sur la représentation en treillis est proposée qui consiste à tronquer le nombre de parcours candidats, et de forcer une décision à une étape k pour le choix des paramètres des blocs qui précèdent l'étape $k - d$. D'après les simulations effectuées, la considération d'un nombre d'étapes plus grand que 6 donne aucun gain appréciable, tandis que la troncature du nombre d'étapes considérées réduit la complexité et surtout le délai associé à cette approche. On note aussi que la considération du coût de codage différentiel des vecteurs réduit l'aspect *aléatoire* du champ estimé qui est souvent observé lorsque l'estimation de mouvement est basée strictement sur la minimisation d'un critère de différence absolue minimale. L'optimisation débit-distorsion a donc un effet naturel de lissage sur les champs de déplacement. Des gains non-négligeables sont rapportés dans des simulations qui comparent l'algorithme proposé au modèle standard TMN8 de H.263.

2.2.8 Prédiction multi-trame

Dans *High-Order Motion Compensation for Low Bit Rate Video* [GWS⁺98, WLG98] on présente une formulation optimale au sens débit distorsion d'un codeur hybride multi-hypothèse où la prédiction se fait à partir de plusieurs trames précédentes plutôt que d'une seule trame comme il est généralement le cas. La sélection du vecteur de déplacement ainsi que la trame précédente à laquelle on fait référence doit donc être déterminé. On propose une formulation lagrangienne qui inclut le coût de codage des vecteurs dans l'algorithme d'estimation. On rapporte des gains de l'ordre de 1.5-2dB pour les séquences *Foreman* et *Mother and Daughter* ou encore une réduction de jusqu'à 30% du débit pour une distorsion équivalente.

Dans *Multiple Reference Picture Video Coding Using Polynomial Motion Models* [WSSG98], plutôt que de considérer plusieurs trames références, on produit plusieurs prédicteurs en utilisant différents modèles de mouvement d'ordre supérieur, plus précisément dans cet article on considère des modèles de compensation affines (6 paramètres). Ces M champs de déplacement polynômiaux sont utilisés pour former M prédicteurs, qui sont ensuite disponibles

pour une prédiction de façon analogue à la compensation multi-trame présenté ci-haut. On présente aussi une méthode pour estimer les paramètres des champs polynômiaux dominants à partir de champs de déplacement denses connus. Cette approche a une certaine similarité avec le codage basé sur un dictionnaire de modèles de mouvement que nous avons proposé en [LD98]. Ces travaux ont été développés indépendamment.

2.2.9 Analyse stochastique de l'efficacité de la compensation en mouvement

Une analyse théorique du codage vidéo par des codeurs hybrides avec compensation en mouvement repose nécessairement sur plusieurs suppositions et simplifications. Pour citer l'auteur de *Efficiency Analysis of Multi-Hypothesis Motion-Compensated Prediction for Video Coding* [Gir00] et des travaux connexes [Gir87, Bus97] ;

Une théorie, même approximative, peut informer notre compréhension et guider nos efforts de développement de codeurs vidéo de pointe.

Et c'est ainsi qu'il présente une analyse très générale de l'efficacité de la compensation en mouvement pour la prédiction dans le codage vidéo. On rappelle d'abord le lien entre la puissance spectrale du résiduel de prédiction et la précision de la compensation en mouvement représentée par une densité de probabilité de l'erreur sur le déplacement réel tels que présentés dans [Gir87]. Le modèle prend aussi en considération une composante de bruit résiduel (la composante du signal qui ne peut être représentée par la compensation en mouvement). Le modèle théorique de bornes débit-distorsion sont maintenant étendus à une formulation plus générale de la compensation multi-hypothèse qui fait la compensation par une combinaison linéaire de plusieurs prédicteurs. La formulation multi-hypothèse généralise simultanément, la prédiction bi-directionnelle (trames B de MPEG), la compensation par blocs superposés (*Overlapped Block Motion Compensation(OBMC)*) et la prédiction multi-frames (*Long term memory motion compensation*).

Dans ce contexte le problème d'estimation des paramètres de prédiction est posé comme un problème de prédiction linéaire, où on tente d'estimer les poids relatifs de plusieurs prédicteurs (Les poids sont en fait déterminés indépendamment à chaque fréquence donc les poids optimaux sont en fait des fonctions de transfert de filtres optimaux)). Il propose une solution optimale débit-distorsion de cette formulation. Les résultats généraux sont ensuite comparés à plusieurs scénarios expérimentaux en indiquant que l'analyse théorique concorde bien avec les comportements expérimentaux observés. Il en découle les conclusions générales suivantes :

- Une combinaison optimale de N hypothèses réduit toujours le taux optimal de codage pour N croissant.
- En présence de bruit résiduel réaliste (modélisation par le mouvement imparfaite) l'augmentation du nombre d'hypothèses produit des gains marginaux décroissants et éventuellement saturés.
- La précision dite critique de la compensation en mouvement (passée laquelle le gain de codage devient négligeable) devient plus grande en présence de bruit résiduel croissant et en présence d'un nombre croissant d'hypothèses, donc la précision sous-pixel de la compensation en mouvement est moins importante en présence d'hypothèses multiples.

2.3 Estimation et compensation en mouvement

Étant donné que l'approche débit-distorsion nous offre des outils pour déterminer les paramètres optimaux d'encodage, nous croyons qu'il sera important que la représentation sous-jacente soit assez riche (flexible) pour permettre un gain substantiel de codage. Pour cette raison nous présentons ici de tels modèles *plus riches* pour la compensation en mouvement. Sans que cette liste soit exhaustive, elle donne un aperçu de représentations possibles du mouvement qu'il serait peut être souhaitable de combiner avec une approche d'optimisation globale débit-distorsion.

Le mouvement sert à relier des positions spatiales de différentes trames. Cette information est souvent utilisée pour effectuer la prédiction d'une image à partir d'une autre. *e.g.* un prédicteur $P(x, y, t)$ de l'image $I(x, y, t)$ pourraient être défini par

$$P(x, y, t) = I(M_i(x, y), t - \delta_t) \quad \forall (x, y) \in B_i \quad (2.5)$$

où la fonction $M_i : x \mapsto x'$ représente le modèle de mouvement pour la région B_i .

Plusieurs formes de M ont été utilisées. Dans leur article *Motion Compensation Based on Spatial Transformations* [NH94] Nakaya et Harashima font un recensement des modèles qui ont été essayés et les classifient ainsi :

Translation pure

$$M_{i,x}(x, y) = x - u_i \quad M_{i,y}(x, y) = y - v_i \quad (2.6)$$

Transformations affines

$$M_{i,x}(x, y) = u_2x + u_1y + u_0 \quad M_{i,y}(x, y) = v_2x + v_1y + v_0 \quad (2.7)$$

Transformations bilinéaires

$$M_{i,x}(x, y) = u_3xy + u_2x + u_1y + u_0 \quad (2.8)$$

$$M_{i,y}(x, y) = v_3xy + v_2x + v_1y + v_0$$

Transformation en perspective

$$M_{i,x}(x, y) = \frac{u_2x + u_1y + u_0}{z_x x + z_y y + 1} \quad (2.9)$$

$$M_{i,y}(x, y) = \frac{v_2x + v_1y + v_0}{z_x x + z_y y + 1}$$

Et on pourrait ajouter le modèle suivant de [PC95].

Transformations géométriques de second ordre

$$M_{i,x}(x, y) = u_5x^2 + u_4y^2 + u_3xy + u_2x + u_1y + u_0 \quad (2.10)$$

$$M_{i,y}(x, y) = v_5x^2 + v_4y^2 + v_3xy + v_2x + v_1y + v_0$$

Un modèle de transformations affines adaptées à une structure de blocs hiérarchiques est présenté dans [Hua94]. Dans la thèse [Foo95] on introduit le concept plus raffiné de trajectoires de mouvement.

Ces modèles sont tous très semblables et offrent simplement une généralisation progressive (dans l'ordre où ils sont présentés) du même modèle. Ces articles, en plus d'introduire les modèles de mouvement, proposent des techniques d'estimation des paramètres de ces modèles et on s'attarde souvent à la complexité de ces algorithmes. Ces aspects dépassent le cadre de cet exposé.

Pour une revue générale et récente des techniques d'estimation et de compensation en mouvement nous nous sommes référés à l'article [SK99].

Chapitre 3

Optimisation débit-distorsion

Nous essayons dans cette section de présenter une formulation générale pour la famille de problèmes que nous voulons résoudre. Nous verrons que le modèle général doit être adapté pour être utile dans le contexte d'un codec spécifique.

Soit $\mathbf{p} = (p_i)$ un vecteur de paramètres (variables indépendantes) et les fonctions $R(\mathbf{p})$ et $D(\mathbf{p})$ représentant respectivement le débit et la distorsion associés au paramètre \mathbf{p} .

On peut substituer au problème

$$\min_{\mathbf{p}} D(\mathbf{p}) \quad \text{s.c.} \quad R(\mathbf{p}) \leq R_0 \quad (3.1)$$

un problème équivalent de minimisation sans contrainte :

$$\min_{\mathbf{p}} J_\lambda(\mathbf{p}) \quad \lambda > 0, \quad (3.2)$$

avec $J_\lambda(\mathbf{p}) = D(\mathbf{p}) + \lambda R(\mathbf{p})$. Cette formulation relaxée donne la même solution \mathbf{p}^* par la méthode des multiplicateurs de Lagrange. Pour être plus précis on dit que s'il existe un $\lambda^* > 0$ tel que \mathbf{p}^* soit une solution au problème (3.2) et $R(\mathbf{p}^*) = R_0$, alors \mathbf{p}^* est aussi une solution au problème original (3.1). Tout le développement qui est fait est aussi transférable au problème dual de minimisation sujet à une contrainte sur la distorsion

$$\min_{\mathbf{p}} R(\mathbf{p}) \quad \text{s.c.} \quad D(\mathbf{p}) \leq D_0, \quad (3.3)$$

par un simple échange des quantités $R(\mathbf{p})$ et $D(\mathbf{p})$.

Nous dérivons ainsi les conditions nécessaires suivantes pour l'obtention d'un minimum local (en \mathbf{p}) :

$$\frac{\partial}{\partial p_i} J_\lambda = \frac{\partial}{\partial p_i} D + \lambda \frac{\partial}{\partial p_i} R = 0 \quad \forall i \quad (3.4)$$

Pour des problèmes plus concrets, les paramètres (\mathbf{p}) doivent être identifiés à des quantités plus précises. Un des défis est de trouver une manière efficace de calculer les quantités $\frac{\partial}{\partial p_i} D$ et $\frac{\partial}{\partial p_i} R$.

Cette formulation est sensiblement différente de l'approche de [Gir94] où le but est d'effectuer une partition d'un débit total R entre la composante mouvement (R_d) et la composante résiduel d'image (R_e). On doit donc estimer directement $\frac{\partial D}{\partial R_e}$ et $\frac{\partial D}{\partial R_d}$. Nous préférons notre formulation puisque les expressions pour le débit et la distorsion sont considérés indépendamment même si les deux quantités dépendent à leur tour de paramètres communs. Ces conditions s'expriment aussi par l'idée que la contribution marginale en réduction de distorsion par une variation d'un paramètre p_i devrait avoir la même *efficacité* en termes de coût de codage pour tous les paramètres p_i .

La formulation en termes de débit-distorsion nous apportera deux éléments.

- La caractérisation de la performance du codec pour un grand ensemble de paramètres à la fois, et même la comparaison directe de codecs qui opèrent sur un ensemble de paramètres très différents.
- Une formulation mieux adaptée à la solution du problème de minimisation.

3.1 Méthode des multiplicateurs de Lagrange

Dans plusieurs des problèmes que nous examinerons, et pour les fins de cet exposé, il est utile pour nous de caractériser complètement la courbe débit-distorsion, i.e. de présenter d'un seul coup, l'ensemble des points (R, D) réalisables par un codec. Pour ce portrait complet il nous faut essentiellement faire un essai exhaustif de combinaisons de paramètres \mathbf{p} .

En pratique, l'évaluation de $R(\mathbf{p})$ et $D(\mathbf{p})$ est coûteux (temps de calcul, mémoire, ...). De plus la combinatoire des paramètres admissibles peut croître très rapidement. Nous voulons donc minimiser le nombre d'évaluations de ces quantités. Dans un contexte opérationnel donc, la formulation lagrangienne nous laisse deux problèmes ; comment trouver λ^* , et comment trouver \mathbf{p}^* pour ce λ^* .

Tel que nous verrons avec l'exemple qui suit, notre première option est de faire une recherche exhaustive des paramètres pour complètement caractériser l'enveloppe convexe. La deuxième option est de faire une recherche diadique sur λ en essayant de produire le λ^* désiré en un nombre minimum d'étapes.

Pour faire une recherche efficace de l'espace λ , nous accumulons tous les points (R, D) réalisables pour un problème dans une liste. Nous choisissons ensuite les points qui constituent son enveloppe convexe. Nous modélisons ensuite cette enveloppe convexe d'une manière à pouvoir produire un bon candidat pour λ^* .

$$\mathcal{R} = (R, D)_i \tag{3.5}$$

$$\text{Conv}(\mathcal{R}) = \{(R, D)_i \mid \exists \lambda \text{ t.q. } D_i + \lambda R_i \leq D_j + \lambda R_j, \forall j \neq i\} \tag{3.6}$$

Identifions cette notation à un exemple plus proche du codage vidéo.

3.2 Choix de λ

Nous pouvons modéliser l'enveloppe convexe par un modèle de la courbe

$$D(Q) = \sum_j a_j g_j(R) \text{ avec } g_j(R) \in \{R^0, R^{-1}, R^{-2}\} \tag{3.7}$$

ou encore un modèle linéaire ou bezier par morceaux. Ces modèles produisent des modèles complets de la courbe R, D optimale, de laquelle nous pouvons estimer λ^* . Cependant il est plus simple de considérer le modèle de l'enveloppe convexe seulement sur un intervalle de recherche. λ^* est estimé par une séquence convergente $\lambda_1, \lambda_2, \lambda_3, \dots$

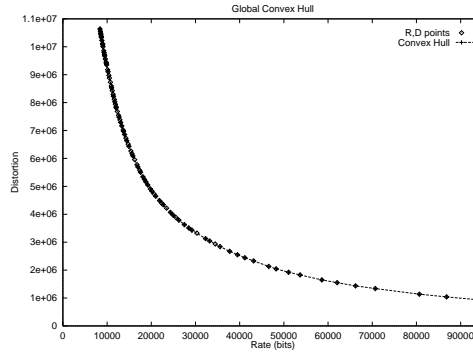


FIG. 3.1 – Estimés successif de λ

Le graphe devrait montrer le précuss du choix de lambda.

Comparaison des vitesses de convergence des différents modèles de $R(D)$ comme dans [SK98] p.1511, mais avec des données empiriques.

3.3 Problème simplifié : codage d'une image

Voici un problème qui suit la forme du modèle décrit ci-haut. Il n'adresse pas la question d'allocation du débit à la représentation du mouvement, il sert seulement à démontrer la solution d'un problème simplifié par l'approche des multiplicateurs de Lagrange.

On désire coder une image par la méthode suivante qui suit grossièrement le codage des images *intra* dans un codeur MPEG.

- Division de l'image en blocs de 8 x 8.
- Transformation par DCT de chaque bloc.
- Quantification du bloc i par un facteur de quantification q_i . En fait chaque coefficient de la DCT est quantifié de manière uniforme et le pas de quantification d'un coefficient est obtenu par la multiplication d'une constante associée à ce coefficient par le facteur de quantification du bloc i
- Les coefficients de la DCT quantifiés sont ensuite codés par un codeur entropique par plages.

Soient les coefficient de quantification q_i associés au bloc $B_i = \{x | x \in bloc\ i\}$ et une

image $I(x)$. En supposant que la contribution de chaque bloc au débit et à la distorsion sont indépendants, nous obtenons :

$$D(q) = \sum_i D_i(q_i), \quad R(q) = \sum_i R_i(q_i)$$

$$D_i(q_i) = \sum_{x \in B_i} [I(x) - \text{DCT}^{-1}(Q(\text{DCT}(I(x)), q_i))]^2$$

et

$$R_i(q_i) = R^{(q)}(q_i) + R^{(I)}(q_i)$$

Ici $R^{(I)}(q_i)$ représente le débit pour coder le bloc lui même avec un codage entropique par plages, tandis que $R^{(q)}(q_i)$ représente le coût de coder le facteur de quantification q_i .

Nous examinons ici deux cas, le premier où les q_i sont tous identiques, i.e. fixe pour tous les blocs, et le deuxième où on trouve un facteur q_i distinct par bloc. Et nous verrons par la suite deux approches différentes pour la modélisation de R et D en fonction des paramètres p , une modélisation basé sur un estimé par moindres carrés et un modèle basé sur l'enveloppe convexe des points R, D .

Sous des hypothèses de convexité et de différentiabilité nous pourrions résoudre ce problème. Dans une implantation concrète par contre il n'est pas évident d'évaluer les dérivés partielles nécessaires. Nous regardons d'abord un modèle paramétrique des quantités R et D en fonction des q_i , représentées par une combinaison linéaire de quelques fonctions de base, permettant ainsi d'approximer d'une manière raisonnable les dérivés partielles de ces quantités.

3.3.1 Facteur de quantification fixe

Nous pouvons dans ce cas identifier le coefficient unique $q = q_i \forall i$ au paramètre p du problème initial. Et les conditions nécessaires deviennent :

$$\frac{\partial}{\partial q} D + \lambda \frac{\partial}{\partial q} R = 0$$

Dans ce cas il n'y a qu'un seul modèle pour $R(q)$ et $D(q)$

3.3.2 Facteur de quantification variable

Nous pouvons dans ce cas identifier les coefficients q_i aux paramètres p du problème initial. Et les conditions nécessaires deviennent :

$$\frac{\partial}{\partial q_i} D_i + \lambda \frac{\partial}{\partial q_i} R_i = 0 \quad \forall i$$

Le facteur de quantification de chaque bloc doit aussi être encodé. Pour cet exemple simple les q_i sont bêtement encodés par un code de 6 bits (pour les 64 valeurs admissibles).

Dans ce cas il y a un modèle pour chaque bloc dans l'image.

3.3.3 Modélisation par moindres carrés

Pour chaque bloc nous représentons $R_i(q_i)$ et $D_i(q_i)$ par un modèle simple :

$$R_i(q_i) = \sum_j a_{i,j}^R g_j^R(q_i) \quad \text{et} \quad D_i(q_i) = \sum_j a_{i,j}^D g_j^D(q_i)$$

Les coefficients $a_{i,j}$ sont estimés par une méthode de moindres carrés avec des données qui sont directement évalués : on calcule directement le débit et la distorsion de chaque bloc pour quelques valeurs de q_i .

Ces modèles nous permettent d'évaluer simplement les dérivées partielles de R_i et D_i .

Nous avons essayé (entre autres) les modèles suivants :

$$\{g_j^R(q)\} = \{q^0, q^{-1}, q^{-2}\}$$

$$\{g_j^D(q)\} = \{q^0, q^1, q^2\}$$

Pour résoudre le problème de minimisation, il s'agit de déterminer (pour un λ donné) les q_i qui satisfont :

$$\frac{\partial}{\partial q_i} D_i(q_i) + \lambda \frac{\partial}{\partial q_i} R_i(q_i) = 0$$

Nous obtenons la solution numérique à ce problème par la recherche de zéro par la méthode de Newton.

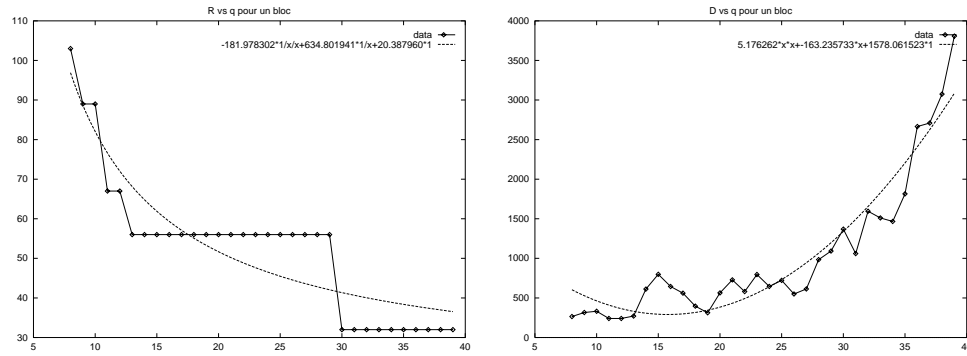


FIG. 3.2 – Courbe débit et distorsion versus facteur de quantification pour un bloc typique

Nous avons essayé différentes fonctions de bases pour le modèle et quelques remarques s'imposent. La qualité du modèle dépend de plusieurs facteurs. Premièrement il doit être une approximation raisonnable des valeurs directement observées, sinon la solution (q_i) ne sera pas très bonne. Deuxièmement le modèle choisi doit se comporter d'une façon raisonnable pour l'obtention numérique de la solution aux conditions nécessaires : il serait souhaitable que la courbe soit strictement décroissante (croissante) et qu'elle soit convexe au moins sur l'intervalle de valeurs qui nous intéressent.

En pratique nous avons rencontré des difficultés à utiliser cette approche ce qui nous a amené à considérer une approche plus simple : la modélisation directe de l'enveloppe convexe des paramètres réalisables.

3.3.4 Modélisation par l'enveloppe convexe

Dans cette approche on représente l'enveloppe convexe d'un ensemble de points R, D réalisables. Ici il s'agit des valeurs R, D obtenues par la sélection des différents facteurs de quantification. Il est ensuite très simple de trouver le point sur l'enveloppe convexe qui minimise l'expression $D(q) + \lambda R(q)$ pour un λ donné.

L'approche est la même que le facteur de quantification soit fixe ou variable. Dans le premier cas on génère seulement un modèle globale de l'enveloppe convexe, et dans le deuxième on produit un modèle de l'enveloppe convexe pour chaque bloc.

La figure 3.4 représente les courbes débit-distorsion pour deux cas. Le premier cas où

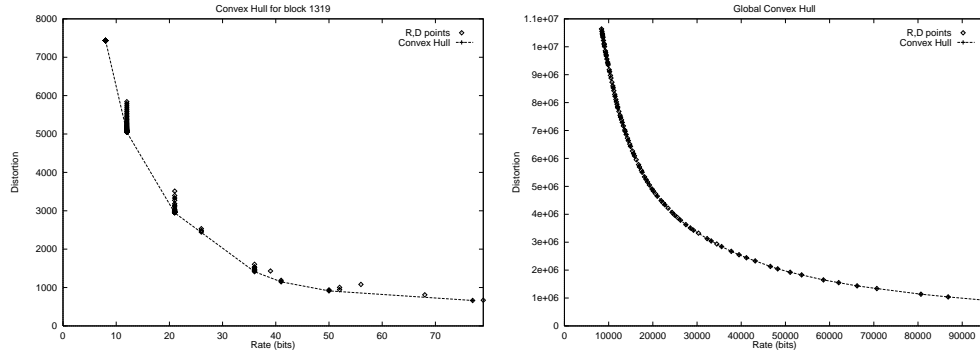


FIG. 3.3 – Enveloppe convexe des points R,D réalisables pour un bloc typique avec q_i variable, et le modèle globale avec q fixe

le facteur de quantification est fixé pour tous les blocs, et le deuxième où le facteur de quantification est variable entre les blocs. Chaque point sur ces enveloppes représente une solution optimale au problème de minimisation avec une différente valeur de λ . Il y donc une valeur de q (ou un ensemble de valeurs q_i) associés à chacun de ces points. On peut voir dans la figure 3.4 qu'il est plus efficace au sens débit-distorsion d'utiliser un facteur de quantification globale pour l'image. Ce résultat dépend cependant de la méthode utilisée pour encoder les différents q_i dans le cas de la quantification variable.

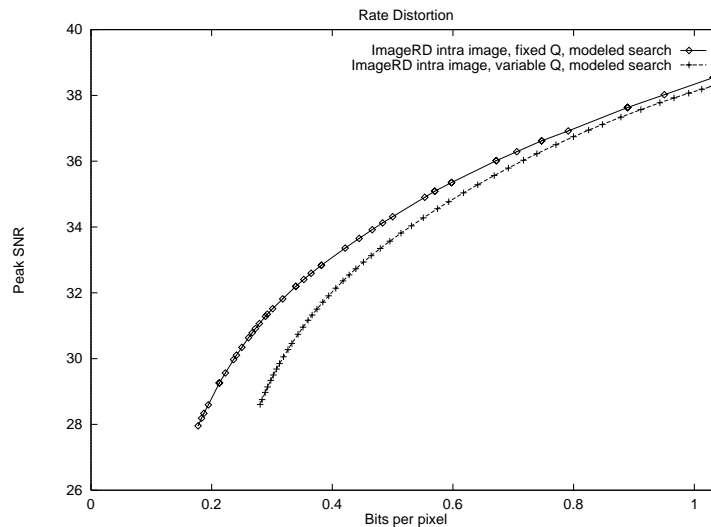


FIG. 3.4 – Courbes débit-distorsion pour q_i variable et q_i fixe

Nous avons déterminé que cette deuxième méthode de modélisation, celle où le facteur

de quantification est fixe, est plus robuste et plus efficace.

3.4 La pratique

Dans le modèle que nous avons présenté il y a un nombre de suppositions qui opèrent. Les courbes que nous considérons sont supposées continues et différentiables. Et pour l'application de la méthode simple de recherche du minimum, on suppose que $D(R)$ est une fonction strictement décroissante et convexe. En pratique les choses ne sont pas si simples. Le premier problème est que la plupart des paramètres considérés sont en fait discrets, que ce soit la taille des blocs, ou les facteurs de quantification. On palie à ce problème en faisant des modèles continus de ces quantités, mais à petite échelle cela peut causer des problèmes.

Il se peut aussi qu'en pratique, il se produise des aberrations qui violent nos suppositions de décroissance uniforme, ou de convexité.

De plus, il arrive souvent que la solution du problème de minimisation du lagrangien pour un λ donné résulte en une recherche combinatoire des paramètres discrets. Il faudra donc examiner de près la complexité de la résolution du problème de minimisation du lagrangien avec λ fixé. Une solution à considérer sera encore une fois la modélisation directe des quantités R et D en fonction des paramètres. L'analyse des courbes débit-distorsion pour des données empiriques sera à ce moment d'un grand aide pour nous aider à formuler ces modèles ou même à tirer des conclusions plus générales sur les combinaisons optimales de paramètres.

Chapitre 4

Codage de cartes étiquettes

Cette section adresse le problème de codage d'une source discrète de symboles. Cette source peut-être de dimension un ou deux, mais nous traitons spécifiquement de sources avec mémoire. Le but visé, comme nous le verrons dans le chapitre suivant sera de coder une carte d'étiquettes associé a une représentation du mouvement. L'efficacité du codage des étiquettes dépendra de la qualité du modèle probabiliste de la source ainsi que du choix des paramètres de ces modèles. Dans certains cas le codage des paramètres du modèle eux-mêmes devront être codés efficacement. Pour la discussion qui suit nous supposons que la source d'étiquettes peut être représentée adéquatement par un modèle Markovien causal.

4.1 Modélisation de la source

Nous désirons coder une séquence de longueur finie (les étiquettes) de longueur M , dont les valeurs proviennent d'un alphabet source \mathcal{A} de taille N . Nommons ces symboles $\mathcal{A} = \{1, 2, \dots, N\}$. La séquence elle, sera représentée par $l = \{l[1], l[2], \dots, l[M]\}$. Dans le cas d'une séquence bi-dimensionnelle, convenons que la séquence est définie par une énumération connue de toutes les positions de l'image. Cet énumération pourrait être un simple balayage par lignes, mais pourrait tout aussi bien être un parcours plus complexe comme un parcours d'Hilbert ou un autre sorte de balayage hiérarchique.

Nous supposons que la source est générée selon un modèle probabiliste connu. Nous considérons plus spécifiquement des modèles qui sont déterminés par un nombre fini de paramètres. Ces paramètres sont représentés par le vecteur θ .

Si la probabilité d'une séquence observée l , selon ce modèle est de $\Pr_{\theta}(l)$, alors un codage efficace, adapté à ce modèle devrait représenter la séquence l avec approximativement $-\log_2(\Pr_{\theta}(l)) + b(\theta)$ bits, où $b(\theta)$ est le nombre de bits requis pour représenter le vecteur paramètre spécifiques à la source observée.

Nous pouvons alors exprimer la probabilité totale de la séquence l observée comme

$$\Pr(l) = \prod_{i=1}^M \Pr(l[i] \mid l[1], \dots, l[i-1]). \quad (4.1)$$

Le but du modèle présenté sera de nous permettre de calculer les probabilités conditionnelles des symboles de la séquence étant donné un paramètre θ . Nous supposons de surcroît que ces probabilités conditionnelles dépendront seulement de quelques échantillons précédents de la source que nous désignons *voisinage* d'un échantillon. Ainsi, le voisinage de l'échantillon à la position i est noté $\mathcal{N}_i \subset \{1, 2, \dots, i-1\}$ et nous avons

$$\Pr(l[i] \mid l[1], \dots, l[i-1]) = \Pr(l[i] \mid l[j], j \in \mathcal{N}_i). \quad (4.2)$$

Le modèle associé à un vecteur paramètre nous permet donc de calculer les probabilités conditionnelles requises.

Nous développons maintenant l'expression de la probabilité de la séquence source conditionnelle aux paramètres de quelques modèles choisis. Le développement nous permettra aussi d'estimer les paramètres en question à partir d'une source observée, ce qui en sera l'usage final dans le chapitre suivant.

4.2 Modèle sans mémoire

Le modèle paramétrique le plus simple est le modèle sans mémoire. Dans ce cas, nous supposons que les échantillons successifs sont indépendants, ou de manière équivalente

$\mathcal{N}_i = \emptyset, \forall i$. Bien que ce modèle ne soit pas d'une grande utilité pratique, il nous permet néanmoins d'exposer simplement l'analyse des probabilités conditionnelles.

Si nous supposons un modèle stationnaire, nous obtenons N paramètres $\theta_1, \dots, \theta_N$ tels que

$$\Pr_{\theta}(l[i] = k \mid l[1], \dots, l[i-1]) = \theta_k, \quad (4.3)$$

et donc

$$\Pr_{\theta}(l) = \prod_{i=1}^M \theta_{l[i]}. \quad (4.4)$$

Il suit immédiatement que

$$-\log_2 \Pr_{\theta}(l) = -\sum_{i=1}^M \log_2 \theta_{l[i]}. \quad (4.5)$$

Définissons la fonction indicateur

$$\delta_m(k) = \begin{cases} 1 & k = m \\ 0 & k \neq m. \end{cases} \quad (4.6)$$

Soit maintenant

$$q_m = \sum_{i=1}^M \delta_m(l[i]), \quad (4.7)$$

le nombre d'échantillons de la séquence l ayant la valeur m . On a aussi que $\sum_{m=1}^N q_m = M$.

Nous pouvons donc écrire

$$\begin{aligned} -\log_2 \Pr_{\theta}(l) &= -\sum_{i=1}^M \sum_{m=1}^N \log_2(\theta_m) \delta_m(l[i]) \\ &= -\sum_{m=1}^N \log_2(\theta_m) \sum_{i=1}^M \delta_m(l[i]) \\ &= -\sum_{m=1}^N q_m \log_2(\theta_m). \end{aligned} \quad (4.8)$$

Ainsi le nombre total de bits requis pour coder la séquence l étant donné le vecteur paramètre θ sera

$$R_{\theta} = - \sum_{m=1}^N q_m \log_2(\theta_m) + b(\theta). \quad (4.9)$$

Nous désirons maintenant minimiser ce coût de codage pour une séquence observée par un choix approprié (optimal) de θ .

Considérons d'abord la minimisation de $-\sum_{m=1}^N q_m \log_2(\theta_m)$ en supposant que la contribution de $b(\theta)$ sera négligeable.

De l'inégalité de l'information, nous avons

$$\begin{aligned} -\log_2 \Pr_{\theta}(l) &= M \sum_{m=1}^N \frac{q_m}{M} \log_2 \left(\frac{1}{\theta_m} \right) \\ &\geq M \sum_{m=1}^N \frac{q_m}{M} \log_2 \left(\frac{1}{q_m/M} \right) \end{aligned} \quad (4.10)$$

avec égalité si et seulement si $\theta_m = q_m/M, m = 1, \dots, N$. Nous obtenons donc que le choix optimal des paramètres θ qui minimise le coût de codage de la séquence, étant donné que ces mêmes paramètres sont connus, sont simplement les fréquence d'apparition des N symboles, ou une distribution de probabilité que nous identifions à \mathbf{q} . Le nombre de bits requis est M fois l'entropie de cette distribution de probabilités.

Nous pouvons donc complètement spécifier le modèle en transmettant les valeurs q_m , desquelles on peut retrouver le θ . Le coût de codage de ces paramètres avec un code très simple serait de $N \lceil \log_2 M \rceil$ bits étant donné que tous les q_m sont des entiers entre 1 et M .

Illustrons tout ceci avec un exemple simple. Soit une image de 512×512 points ($M = 262144$) et un alphabet de taille $N = 30$, avec une entropie de 0.1 bits/symbole ou environ 26214 bits. Le coût de codage des paramètre serait de $30 \times 18 = 540$ bits, soit 0.002 bits/symbole ou 2% du coût total.

Si nous utilisons une approximation $\tilde{\mathbf{q}}$ du vecteur paramètre optimal \mathbf{q} , le coût de codage augmentera de $D(\mathbf{q}||\tilde{\mathbf{q}})$ où

$$D(\mathbf{p}||\mathbf{q}) = \sum_{i=1}^N p_i \log_2 \left(\frac{p_i}{q_i} \right). \quad (4.11)$$

Par exemple, l'approximation de q par une quantification à 14 bits plutôt qu'à 18 bits dans un exemple numérique correspondant aux chiffres ci-haut, a réduit le coût total R_{θ} de 65 bits, tandis qu'une quantification à 13 bits, l'a augmenté de 178 bits.

4.3 Modèle de premier ordre

Par un modèle de premier ordre nous entendons un modèle dans lequel

$$\Pr_{\theta}(l[i] \mid l[1], \dots, l[i-1]) = \Pr_{\theta}(l[i] \mid l[i-1]) \quad \forall i. \quad (4.12)$$

4.3.1 Probabilités de transition complètes

Le modèle de premier ordre le plus simple, en est un où l'on spécifie toutes les probabilités de transition :

$$\Pr_{\theta}(l[i] = m \mid l[i-1] = j) = \theta_{jm}, \quad (4.13)$$

avec $\sum_{m=1}^N \theta_{jm} = 1$ pour que le modèle soit consistant, et que les probabilités conditionnelles $\{\theta_{j1}, \dots, \theta_{jN}\}$ forment une distribution de probabilité valide. Ainsi ce modèle a $N-1$ paramètres libres pour chaque j et $N(N-1)$ paramètres au total. Le coût de codage d'une séquence l avec ce modèle serait de

$$R_{\theta} = - \sum_{i=1}^M \log_2 \Pr_{\theta}(l[i] \mid l[i-1]) + b(\theta), \quad (4.14)$$

où une valeur appropriée (connue) serait substituée à $l[0]$. Comme avant, le coût de codage de la séquence (excluant le coût de codage des paramètres eux-mêmes) peut s'écrire

$$\begin{aligned}
-\log_2 \Pr_{\theta}(l) &= -\sum_{i=1}^M \log_2 \theta_{l[i-1],l[i]} \\
&= -\sum_{i=1}^M \sum_{j=1}^N \sum_{m=1}^N \log_2 \theta_{jm} \delta_j(l[i-1]) \delta_m(l[i]) \\
&= -\sum_{j=1}^N \sum_{m=1}^N \log_2 \theta_{jm} \sum_{i=1}^M \delta_j(l[i-1]) \delta_m(l[i]) \\
&= -\sum_{j=1}^N \sum_{m=1}^N r_{jm} \log_2 \theta_{jm}
\end{aligned} \tag{4.15}$$

où

$$r_{jm} = \sum_{i=1}^M \delta_j(l[i-1]) \delta_m(l[i]) \tag{4.16}$$

est le nombre d'échantillons dans l avec une valeur m et dont l'échantillon précédent a la valeur j . Définissons

$$q_j = \sum_{m=1}^N r_{jm}, \tag{4.17}$$

le nombre total d'échantillons de l dont l'échantillon précédent a la valeur j . Ainsi les vecteurs $\mathbf{r}_j = \{r_{jm}/q_j, m = 1, \dots, N\}$ forment de distributions de probabilité marginales bien formées pour chaque j . Avec cette notation le coût de codage de la séquence devient

$$\begin{aligned}
-\log_2 \Pr_{\theta}(l) &= \sum_{j=1}^N q_j \sum_{m=1}^N \frac{r_{jm}}{q_j} \log_2 \frac{1}{\theta_{jm}} \\
&\geq \sum_{j=1}^N q_j \sum_{m=1}^N \frac{r_{jm}}{q_j} \log_2 \frac{1}{r_{jm}/q_j}
\end{aligned} \tag{4.18}$$

ou l'égalité est satisfaite si et seulement si $\theta_{jm} = r_{jm}/q_j$. On peut voir ce résultat comme un raffinement du développement fait pour le modèle sans mémoire et utilisé sur les sous-séquences d'échantillons qui ont un prédécesseur $l[i-1]$ de valeur identique. Cependant le coût de codage de la séquence, si on utilise le même code que dans la section précédente

pour coder les paramètres θ ou r_{jm} serait d'environ $b(\theta) = N^2 \lceil \log_2 M \rceil$ bits, qui est moins négligeable qu'à l'exemple précédent. De fait pour l'exemple précédent le coût de codage représenterait maintenant environ 40% du coût de codage de la séquence. Ainsi il serait beaucoup plus important dans ce cas de trouver un codage plus efficace des paramètres.

Nous préférons par contre, aller vers un modèle avec un nombre réduit de paramètres.

4.3.2 Modèle à nombre de paramètres réduit

On pourrait imaginer bon nombre de modèles plus simples qui captent une partie de la dépendance entre les échantillons successifs. Nous présentons ici un tel modèle. Supposons que la probabilité que deux échantillons soient identiques pour toutes les valeurs d'échantillons, i.e.,

$$\Pr_{\theta}(l[i] = j \mid l[i-1] = j) = \theta_s, \quad j = 1, \dots, N. \quad (4.19)$$

De plus supposons que les autres probabilités marginales pour un événement conditionnant ($l[i-1] = j$) soient proportionnelles à un ensemble de paramètres fixes, i.e.,

$$\Pr_{\theta}(l[i] = m \mid l[i-1] = j) = \alpha_j \theta_m, \quad m \neq j. \quad (4.20)$$

Pour que les probabilités soient bien formés il faudrait alors $\sum_{m=1}^N \Pr_{\theta}(l[i] = m \mid l[i-1] = j) = 1$ pour chaque j et donc

$$\alpha_j = \frac{1 - \theta_s}{\sum_{\substack{m=1 \\ m \neq j}}^N \theta_m}. \quad (4.21)$$

Le vecteur paramètre est donc $\theta_+ = [\theta_1, \dots, \theta_N, \theta_s] = [\theta, \theta_s]$. Comme précédemment,

$$\begin{aligned} -\log_2 \Pr_{\theta}(l) &= -\sum_{j=1}^N \sum_{m=1}^N r_{jm} \log_2 \Pr_{\theta}(l[i] = m \mid l[i-1] = j) \\ &= -\sum_{j=1}^N \left(\sum_{\substack{m=1 \\ m \neq j}}^N r_{jm} \log_2 \frac{(1 - \theta_s) \theta_m}{\sum_{\substack{l=1 \\ l \neq j}}^N \theta_l} + r_{jj} \log_2 \theta_s \right). \end{aligned} \quad (4.22)$$

Nous voulons encore une fois minimiser cette expression par un choix approprié de $\boldsymbol{\theta}_+$ en respectant les contraintes $\theta_i \geq 0$ pour tout i et $0 \leq \theta_s \leq 1$. Nous pouvons réécrire l'expression précédente comme

$$\begin{aligned}
-\log_2 \Pr_{\boldsymbol{\theta}}(l) &= -\log_2(1 - \theta_s) \left(\sum_{j=1}^N \sum_{\substack{m=1 \\ m \neq j}}^N r_{jm} \right) - \log_2 \theta_s \left(\sum_{j=1}^N r_{jj} \right) & (A) \\
&+ \sum_{j=1}^N \log_2 \left(\sum_{\substack{l=1 \\ l \neq j}}^N \theta_l \right) \left(\sum_{\substack{m=1 \\ m \neq j}}^N r_{jm} \right) - \sum_{j=1}^N \sum_{\substack{m=1 \\ m \neq j}}^N r_{jm} \log_2 \theta_m. & (B)
\end{aligned} \tag{4.23}$$

Nous pouvons minimiser cette expression en deux parties. La ligne (A) peut être minimisée par le choix de θ_s , tandis que la ligne (B) peut être minimisée par le choix des autres paramètres de $\boldsymbol{\theta}_+$ soient $\boldsymbol{\theta}$. L'expression en (A) est minimisée par

$$\theta_s = \frac{\sum_{j=1}^N r_{jj}}{M}. \tag{4.24}$$

L'ensemble des valeurs de $\boldsymbol{\theta}$ qui minimisent (B) n'est cependant pas unique puisque la multiplication des composantes θ_i du vecteur par une constante t , préserve la valeur de (B) :

$$\log_2 \left(\frac{(1 - \theta_s)t\theta_m}{\sum_{\substack{l=1 \\ l \neq j}}^N t\theta_l} \right) = \log_2 \left(\frac{(1 - \theta_s)\theta_m}{\sum_{\substack{l=1 \\ l \neq j}}^N \theta_l} \right). \tag{4.25}$$

Sans perte de généralité, nous imposons donc une contrainte supplémentaire

$$\sum_{l=1}^N \theta_l = 1. \tag{4.26}$$

Définissons aussi la région (simplexe)

$$\mathcal{B} = \left\{ \boldsymbol{\theta} \mid \sum_{l=1}^N \theta_l = 1, \theta_l \geq 0, l = 1, \dots, N \right\}. \tag{4.27}$$

Notons d'emblée que \mathcal{B} est un sous-ensemble convexe de \mathbb{R}^N . Définissons aussi l'expression de la ligne (B) de l'équation (4.23) comme $\mathcal{F}(\boldsymbol{\theta})$. Il nous reste maintenant à trouver

un θ qui corresponde à $\min_{\theta \in \mathcal{B}} \mathcal{F}(\theta)$. Et nous avons que dans \mathcal{B} ,

$$\sum_{\substack{l=1 \\ l \neq j}}^N \theta_l = 1 - \theta_j \quad \forall j. \quad (4.28)$$

Donc

$$\begin{aligned} \mathcal{F}(\theta) &= \sum_{j=1}^N \log_2(1 - \theta_j) \left(\sum_{\substack{m=1 \\ m \neq j}}^N r_{jm} \right) - \sum_{j=1}^N \sum_{\substack{m=1 \\ m \neq j}}^N r_{jm} \log_2 \theta_m \\ &= \sum_{j=1}^N \sum_{\substack{m=1 \\ m \neq j}}^N r_{jm} \log_2 \left(\frac{1 - \theta_j}{\theta_m} \right) \quad \theta \in \mathcal{B}. \end{aligned} \quad (4.29)$$

Lorsque les composantes θ_j tendent vers 0 ou 1, $\mathcal{F}(\theta)$ tend vers $+\infty$ et donc $\mathcal{F}(\theta)$ doit atteindre son minimum à l'intérieur de \mathcal{B} . Nous transformons le problème de minimiser $\mathcal{F}(\theta)$ sujet à la contrainte que $\theta \in \mathcal{B}$ ou $\sum_{m=1}^N \theta_m = 1$ en utilisant une relaxation lagrangienne :

$$E_\lambda = \sum_{j=1}^N \log_2(1 - \theta_j) \left(\sum_{\substack{m=1 \\ m \neq j}}^N r_{jm} \right) - \sum_{j=1}^N \sum_{\substack{m=1 \\ m \neq j}}^N r_{jm} \log_2 \theta_m + \lambda \sum_{m=1}^N \theta_m. \quad (4.30)$$

Les conditions nécessaires à la minimisation à l'intérieur de \mathcal{B} sont

$$\frac{\partial E_\lambda}{\partial \theta_k} = \frac{-1}{(1 - \theta_k) \ln 2} \sum_{\substack{m=1 \\ m \neq k}}^N r_{km} - \sum_{\substack{j=1 \\ j \neq k}}^N r_{jk} \frac{1}{\theta_k \ln 2} + \lambda = 0. \quad (4.31)$$

En multipliant par $\theta_k(1 - \theta_k)$ et en réécrivant nous obtenons une expression quadratique en θ_k .

$$\theta_k^2 - \theta_k \left(1 + \frac{1}{\lambda \ln 2} \sum_{\substack{m=1 \\ m \neq k}}^N (r_{mk} - r_{km}) \right) + \frac{1}{\lambda \ln 2} \sum_{\substack{m=1 \\ m \neq k}}^N r_{mk} = 0. \quad (4.32)$$

Rappelons nous que d'après l'équation (4.17),

$$\sum_{\substack{m=1 \\ m \neq k}}^N r_{km} = q_k - r_{kk}. \quad (4.33)$$

De manière semblable définissons maintenant

$$q'_k = \sum_{m=1}^N r_{mk}, \quad (4.34)$$

le nombre d'échantillons de l ayant la valeur k . On peut voir que q'_k et q_k diffèrent au plus de 1, due aux premiers et derniers échantillons de la séquence. Nous avons donc

$$\sum_{\substack{m=1 \\ m \neq k}}^N r_{mk} = q'_k - r_{kk}. \quad (4.35)$$

De plus, en posant $\lambda' = \lambda \ln 2$, l'équation (4.32) devient

$$\theta_k^2 - \theta_k \left(1 + \frac{q'_k - q_k}{\lambda'} \right) + \frac{q'_k - r_{kk}}{\lambda'} = 0, \quad (4.36)$$

qui a deux solutions possibles. Nous résolvons ces équations pour obtenir θ_k en fonction de λ' , et nous choisissons ensuite λ' pour satisfaire la contrainte $\sum_{k=1}^N \theta_k = 1$.

Solution exacte

Pour obtenir une solution exacte des θ_k en fonction de λ , et en tenant compte de la contrainte $\sum_{k=1}^N \theta_k = 1$, il nous faut considérer les N^2 combinaisons de racines aux équations quadratiques (4.32,4.36). La figure 4.3.2 illustre avec un exemple numérique,

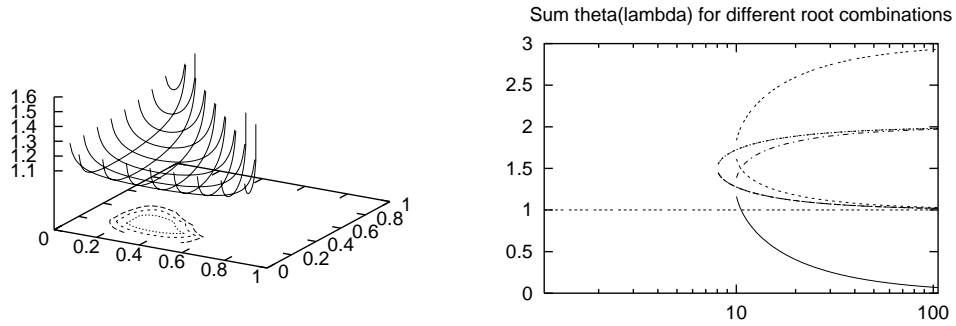


FIG. 4.1 – $R_\theta - b(\theta)$ et $\sum_{k=1}^N \theta_k$ en fonction de λ pour un exemple numérique

où $N = 3, M = 30$, la fonction de coût que nous essayons de minimiser. On y voit $R_\theta - b(\theta)$ en fonction de θ_1, θ_2 (implicitement $\theta_3 = 1 - \theta_1 - \theta_2$). Le deuxième graphe

démontre $\sum_{k=1}^N \theta_k$ en fonction de λ pour les différentes racines admissibles θ_k . On voit que la solution optimale correspond à $\lambda = 10.58$ et $\boldsymbol{\theta} = [0.326, 0.253, 0.421]$.

Nous verrons aussi au chapitre des simulations comment se comparent l'efficacité du codage avec la solution exacte, et la solution approximative que nous présentons maintenant.

Solution approximative

Pour M suffisamment grand, on peut supposer que $(q'_k - q_k)/\lambda' \ll 1$ puisque $|q'_k - q_k| \leq 1$ et que λ' croît normalement avec M . Aussi puisque les probabilités de transitions sont normalement petites nous avons $\theta_k^2 \ll \theta_k$, ce qui nous donne une solution approximative

$$\theta_k \approx \frac{q_k - r_{kk}}{\lambda'}, \quad (4.37)$$

avec $\lambda' \approx \sum_{k=1}^N (q_k - r_{kk})$. Ce qui donne finalement

$$\theta_k \approx \frac{q_k - r_{kk}}{\sum_{k=1}^N (q_k - r_{kk})} \quad (4.38)$$

soit la proportion de tous les points de *transition* dont la valeur de l'échantillon précédent est k . Nous pouvons comparer cette solution approximative à la solution exacte. Nous pouvons transmettre chacun des paramètres de ce modèle avec $\lceil \log_2 M \rceil$ bits, soient $\sum_{j=1}^N r_{jj}$ et les $q_k - r_{kk}, k = 1, \dots, N$ pour un coût total de $b(\boldsymbol{\theta}) = (N + 1)\lceil \log_2 M \rceil$ bits. Ce qui devrait être relativement faible en rapport avec le coût de codage de la séquence elle-même.

Bien que nous puissions concevoir d'autres modèles à paramètres réduits, nous passons tout de suite à des modèles d'ordre supérieur et mieux adaptés au codage d'un champ bi-dimensionnel comme les images étiquettes.

4.4 Modèle de deuxième ordre

Pour les modèles de deuxième ordre, le voisinage de chaque échantillon contient 2 échantillons précédents ($|\mathcal{N}_i| = 2$). Pour des données image ce serait par exemple les

points immédiatement à la gauche et au dessus du point courant. $\mathcal{N}_i = \{i - 1, i - K\}$, pour une image dont la taille des lignes serait de K . Il faudra aussi traiter spécifiquement les points sur la première ligne et la première colonne, puisqu'ils n'aurait autrement qu'un voisin. Pour palier à ce problème, on peut définir une valeur fixe et connue pour tous les échantillons $l[i], i < 0$.

Donc avec ce modèle, nous avons

$$\Pr_{\theta}(l[i] \mid l[1], \dots, l[i - 1]) = \Pr_{\theta}(l[i] \mid l[i - 1], l[i - K]), \quad (4.39)$$

que nous supposons indépendant de i . Encore une fois, nous considérons le modèle avec la pleine matrice de probabilités de transition et nous examinerons ensuite un modèle à paramètres réduits. Bien que le coût de codage des paramètres du modèle a pleine matrice de transition soit prohibitif en pratique, cela fixe quand même une borne inférieure au coût de codage pour tout autre modèle de second ordre au sens de l'équation (4.39).

4.4.1 Probabilités de transition complètes

Dans ce modèle il y a un paramètre pour chaque combinaison des valeurs d'étiquettes pour la position courante et pour chacune des positions du voisinage.

$$\Pr_{\theta}(l[i] = m \mid l[i - 1] = j_1, l[i - K] = j_2) = \theta_{j_1 j_2 m}. \quad (4.40)$$

Nous devons aussi nous assurer que $\sum_{m=1}^N \theta_{j_1 j_2 m} = 1$ pour chacun des (j_1, j_2) , ce qui donne $N^2(N - 1)$ paramètres libres. Comme avant nous écrivons

$$\begin{aligned} -\log_2 \Pr_{\theta}(l) &= -\sum_{i=1}^M \log_2 \theta_{l[i-1], l[i-K], l[i]} \\ &= -\sum_{j_1=1}^N \sum_{j_2=1}^N \sum_{m=1}^N r_{j_1 j_2 m} \log_2 \theta_{j_1 j_2 m} \\ &= \sum_{j_1=1}^N \sum_{j_2=1}^N q_{j_1 j_2} \sum_{m=1}^N \frac{r_{j_1 j_2 m}}{q_{j_1 j_2}} \log_2 \frac{1}{\theta_{j_1 j_2 m}} \\ &\geq \sum_{j_1=1}^N \sum_{j_2=1}^N q_{j_1 j_2} \sum_{m=1}^N \frac{r_{j_1 j_2 m}}{q_{j_1 j_2}} \log_2 \frac{1}{r_{j_1 j_2 m} / q_{j_1 j_2}} \end{aligned} \quad (4.41)$$

avec égalité si et seulement si $\theta_{j_1 j_2 m} = r_{j_1 j_2 m} / q_{j_1 j_2}$, où

$$q_{j_1 j_2} = \sum_{m=1}^N r_{j_1 j_2 m}. \quad (4.42)$$

On peut encore considérer ce résultat comme l'application du résultat du codage sans mémoire aux sous-séquences formées de voisinage conditionnant identique ($l[i-1], \dots, l[i-K]$). Comme au développement du modèle de premier ordre, nous notons que le nombre de paramètres devient rapidement très grand. Avec $N = 30$ cela produirait un espace de 26100 paramètres, ce qui rend un modèle à paramètres réduits impératif.

4.4.2 Modèle à nombre de paramètres réduit

Nous décrivons ici un modèle à paramètres réduits qui a été utilisé avec succès dans le contexte du codage de cartes d'étiquettes chromatiques pour le codage d'images statiques [DFL98], et comme nous le verrons pour le codage de cartes d'étiquettes reliées à des modèles paramétriques de mouvement pour des séquences vidéo.

Dans ce modèle si les deux échantillons du voisinage sont identiques nous supposons que l'échantillon courant prendra aussi cette valeur avec forte probabilité. Nous assignons donc un paramètre (θ_b) à cet événement. Si les valeurs des étiquettes du voisinage sont différents alors nous supposons que l'échantillon courant prendra une de ces deux valeurs avec probabilité égale (θ_e).

Nous supposons de plus que si l'échantillon ne correspond pas à un de ces choix, alors la probabilité que ce soit une des étiquettes restantes sera proportionnelle à un vecteur fixe de paramètres ($\theta_i, i = 1, \dots, N$).

Donc

$$\Pr_{\theta}(l[i] = m \mid l[i-1] = j_1, l[i-K] = j_2) = \begin{cases} \theta_b & \text{si } m = j_1 = j_2, \\ \frac{1-\theta_b}{1-\theta_{j_1}}\theta_j & \text{si } j_1 = j_2 \text{ et } m \neq j_1, \\ \theta_e/2 & \text{si } j_1 \neq j_2 \text{ et } (m = j_1 \text{ ou } m = j_2), \\ \frac{1-\theta_e}{1-\theta_{j_1}-\theta_{j_2}}\theta_j & \text{si } j_1 \neq j_2 \text{ et } m \neq j_1 \text{ et } m \neq j_2. \end{cases} \quad (4.43)$$

Nous pouvons vérifier qu'avec les facteurs de normalisation $(1 - \theta_b)/(1 - \theta_{j_1})$ et $(1 - \theta_e)/(1 - \theta_{j_1} - \theta_{j_2})$ les probabilités conditionnelles sont bien formées et

$$\sum_{m=1}^N \Pr_{\theta}(l[i] = m \mid l[i-1] = j_1, l[i-K] = j_2) = 1 \quad (4.44)$$

pour tout les contextes $(j_1, j_2) \in \mathcal{A}^2$.

En tenant compte de la contrainte $\sum_{m=1}^N \theta_m = 1$, il y a $N + 1$ paramètres libres dans ce modèle.

De manière analogue aux sections précédentes, nous avons déterminé

$$\theta_b = \frac{\sum_{j=1}^N r_{jjj}}{M}, \quad (4.45)$$

$$\theta_e = \frac{\sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N r_{ijj} + r_{jij}}{M}, \quad (4.46)$$

et comme approximation

$$\theta_k \approx \frac{\sum_{\substack{i=1 \\ j \neq k}}^N \sum_{\substack{j=1 \\ j \neq i \\ j \neq k}}^N r_{ijk}}{M}, \quad (4.47)$$

4.5 Généralisation à des voisinages élargis

Ayant déjà passé en revue les modèles sans mémoire, de premier ordre et de deuxième ordre, nous tentons ici de faire une généralisation à des modèles dont le voisinage est de

taille arbitraire. Nous considérons donc un voisinage d'ordre arbitraire mais fixe ($|\mathcal{N}_i| = K$)

$$\Pr_{\theta}(l[i] \mid l[1], \dots, l[i-1]) = \Pr_{\theta}(l[i] \mid \{l[k], k \in \mathcal{N}_i\}), \quad (4.48)$$

que nous supposons indépendant de i . Considérons d'abord le modèle avec la pleine matrice de probabilités de transition et nous tenterons ensuite de déterminer un modèle général à paramètres réduits.

4.5.1 Probabilités de transition complètes

Dans ce modèle il y a un paramètre pour chaque combinaison des valeurs d'étiquettes pour la position courante et pour chacune des positions du voisinage.

$$\Pr_{\theta}(l[i] = v_0 \mid l[j_1] = v_1, \dots, l[j_K] = v_K) = \theta_{v_0 v_1 \dots v_K}. \quad (4.49)$$

Pour obtenir des distributions conditionnelles bien formées nous imposons

$$\sum_{m=1}^N \theta_{m v_1 \dots v_K} = 1, \quad \forall (v_1, \dots, v_K) \in \mathcal{A}^K \quad (4.50)$$

Avec ces contraintes, il y a $N^K(N-1)$ paramètres libres dans notre modèle.

Le coût de codage de la séquence étant donné ces paramètres sera

$$\begin{aligned} -\log_2 \Pr_{\theta}(l) &= -\sum_{i=1}^M \log_2 \theta_{l[i]l[j_1] \dots l[j_K]}, \quad \mathcal{N}_i = \{j_1, \dots, j_K\} \\ &= -\sum_{v_1=1}^N \dots \sum_{v_K=1}^N \sum_{v_0=1}^N r_{v_0 v_1 \dots v_K} \log_2 \theta_{v_0 v_1 \dots v_K} \\ &= \sum_{v_1=1}^N \dots \sum_{v_K=1}^N q_{v_1 \dots v_K} \sum_{v_0=1}^N \frac{r_{v_0 v_1 \dots v_K}}{q_{v_1 \dots v_K}} \log_2 \frac{1}{\theta_{v_0 v_1 \dots v_K}} \\ &\geq \sum_{v_1=1}^N \dots \sum_{v_K=1}^N q_{v_1 \dots v_K} \sum_{v_0=1}^N \frac{r_{v_0 v_1 \dots v_K}}{q_{v_1 \dots v_K}} \log_2 \frac{1}{r_{v_0 v_1 \dots v_K} / q_{v_1 \dots v_K}} \end{aligned} \quad (4.51)$$

avec $q_{v_1 \dots v_K} = \sum_{i=1}^N r_{i v_1 \dots v_K}$, et où l'équation atteint l'égalité si et seulement si

$$\theta_{v_0 v_1 \dots v_K} = \frac{r_{v_0 v_1 \dots v_K}}{q_{v_1 \dots v_K}} \quad (4.52)$$

Avec $|\mathcal{N}_i| = K = 4$ et $N = 30$ cela donnerait 23,490,000 de paramètres. Cependant il faut aussi noter qu'un grand nombre de ces paramètres seraient nuls, ce qui devient tout de suite apparent quand le nombre de paramètres ou de contextes dépasse la longueur de la source ! La borne supérieure du nombre de paramètres générés ne serait pas atteinte en pratique, entre autres parce qu'il ne pourra jamais y avoir plus que M contextes différents.

Voyons maintenant une structure généralisée pour une famille de modèles à paramètres réduits.

4.5.2 Modèle à nombre de paramètres réduits

Définissons premièrement la séquence v_i :

v_i , une séquence ordonnée des éléments du voisinage \mathcal{N}_i , trié en ordre décroissant du nombre de fois que chacun des membres du voisinage apparaît, et s'il y a égalité entre la fréquence d'apparition de deux étiquettes du voisinage, un deuxième critère est utilisé, la position du premier de ces échantillons selon un ordre fixé de positions relatives des membres du voisinage \mathcal{N}_i .

v_i^1 , l'échantillon qui apparaît le plus fréquemment dans le voisinage \mathcal{N}_i .

v_i^2 , l'échantillon qui apparaît le deuxième plus fréquemment dans le voisinage \mathcal{N}_i et ainsi de suite.

Transformons ensuite notre alphabet \mathcal{A} en un nombre réduit de symboles, par une transformation qui dépendra de la position de l'échantillon courant. Soit le nouvel alphabet de taille $N' + 1$

$$\mathcal{A}' = \{C_0, C_1, \dots, C_{N'}\}, \quad (4.53)$$

et la transformation $T_i : \mathcal{A} \mapsto \mathcal{A}'$

$$T_i(m) = \begin{cases} C_k & \text{si } m = v_i^k, \\ C_0 & \text{sinon,} \end{cases} \quad (4.54)$$

et finalement pour une notation semblable à celle que nous utilisons jusqu'ici, la séquence transformée

$$l^i[j] = T_i(l[j]). \quad (4.55)$$

Comme à la section 4.5.1, nous pouvons coder la séquence transformée $\{l^i[i]\}$, ce que nous savons faire avec une pleine matrice de transitions qui a maintenant considérablement moins d'états : $(N' + 1)^K(N')$ plutôt que $N^K(N - 1)$. Rappelons nous aussi qu'en pratique nous n'avons pas à transmettre les coefficients nuls, ce qui se produit pour un grand nombre des contextes.

Avec ce codage à cout réduit, nous avons réussi à transmettre la séquence transformée. Nous devons maintenant retrouver la séquence originale par une transformation inverse. La transformation T_i est inversible pour tout les éléments de la séquence $l^i[i] \in \{\mathcal{A}'\}$ sauf C_0 . En fait en définissant

$$T_i^{-1}(C_j) = \{m \in \mathcal{A} \mid T(m) = C_j\} \quad (4.56)$$

nous avons

$$|T_i^{-1}(C_j)| = \begin{cases} 0 & \text{si } \nexists m \in v_i \text{ tel que } T^i(m) = C_j \\ 1 & \text{si } \exists m \in v_i \text{ tel que } T^i(m) = C_j \\ N - \sum_{k=1}^{N'} |T_i^{-1}(C_k)| & \text{si } j = 0, \end{cases} \quad (4.57)$$

Et le modèle devient

$$\begin{aligned} & \Pr_{\theta}(l[i] = v_0 \mid l[j_1] = v_1, \dots, l[j_K] = v_K) \\ &= \frac{\Pr_{\theta}(l^i[i] = c_0 \mid l^i[j_1] = c_1, \dots, l^i[j_K] = c_K)}{|T_i^{-1}(c_0)|} \\ &= \frac{\theta_{c_0 c_1 \dots c_K}}{|T_i^{-1}(c_0)|}. \end{aligned} \quad (4.58)$$

On peut considérer que ce modèle à paramètres réduits utilise une transformation locale et causale qui sert de prédicteur, et qui réduit l'espace symbole. D'autre part comme la transformation n'est pas réversible dans le cas du symbole C_0 , on peut voir le dénominateur de

l'expression à l'équation (4.58), comme un code supplémentaire de $|T_i^{-1}(C_0)|$ symboles équiprobables qui est simplement composé au code avec l'alphabet réduit lorsque le symbole C_0 apparaît.

L'avantage de ce modèle à paramètres réduits est qu'il retient une partie de l'information de similarité, i.e. pour les étiquettes qui apparaissent fréquemment dans le voisinage, et retient surtout la structure spatiale de cette dépendance à l'intérieur du voisinage, et cela indépendamment des valeurs spécifiques qui sont identifiées à C_1, C_2, \dots dans un voisinage spécifique.

4.5.3 Troncation des contextes

Comme nous avons discuté dans le contexte du codeur sans mémoire (4.2), en modifiant le processus de codage des paramètres θ , par une approximation $\tilde{\theta}$, il est possible de réduire le coût de codage des paramètres, sachant que le codage de la séquence elle-même sera moins efficace (puisque le choix de θ est optimal). Cependant, il se peut bien que le coût total $R_{\tilde{\theta}}$ résultant soit plus petit. On a qu'à comparer

$$R_{\theta} = -\log_2 \Pr_{\theta}(l) + b(\theta) \quad \text{et} \quad R_{\tilde{\theta}} = -\log_2 \Pr_{\tilde{\theta}}(l) + b(\tilde{\theta}). \quad (4.59)$$

Un raffinement possible au modèle, en plus d'une quantification des paramètres telle que suggéré dans la section 4.2, est le suivant. Nous pouvons choisir de ne pas transmettre les coefficients associés à un contexte $\theta_{c_0 c_1 \dots c_K} \forall c_0 \in \mathcal{A}'$, en convenant que tous les événements dont le contexte n'a pas été transmis fait partie d'un contexte spécial (C_0, \dots, C_0) pour lequel la transformée T_i sera modifiée à $T_i(m) = C_0$, ce qui a pour effet d'utiliser un code équiprobable $\Pr_{\theta}(l[i] | l[k] = C_0, \forall k \in \mathcal{N}_i) = 1/N$ lorsque un contexte a été *tronqué*. Le choix de transmettre ou non un contexte est fait au sens de la comparaison en 4.59.

4.5.4 Probabilités marginales

Contrairement aux modèles précédents, nous n'avons pas inclus l'idée d'une probabilité marginale (θ_m du modèle de premier et deuxième ordre). Mais nous faisons ici une

extension à notre modèle généralisé qui a un effet semblable. Cette extension est plus simplement comprise lorsqu'on considère le modèle comme un code pour l'alphabet réduit \mathcal{A} auquel est concaténé un deuxième code quand le symbole C_0 apparaît. Dans la formulation générale toutes les probabilités étaient égales :

$$\Pr_{\theta}(l[i] = m \mid l^i[i] = c_0) = \frac{1}{|T_i^{-1}(c_0)|} \quad \forall m \in T_i^{-1}(c_0). \quad (4.60)$$

Nous proposons plutôt que ces probabilités soient proportionnelles à un vecteur fixe :

$$\Pr_{\theta}(l[i] = m \mid l^i[i] = c_0) = \alpha_i \theta_m \quad \forall m \in T_i^{-1}(c_0), \quad (4.61)$$

avec $\alpha_i = 1/(\sum_{m \in T_i^{-1}(c_0)} \theta_m)$.

$$\Pr_{\theta}(l[i] = v_0 \mid l[j_1] = v_1, \dots, l[j_K] = v_K) = \begin{cases} \theta_{c_0 c_1 \dots c_K} & \text{si } c_0 \neq C_0 \\ \theta_{c_0 c_1 \dots c_K} \alpha_i \theta_{v_0} & \text{si } c_0 = C_0 \end{cases} \quad (4.62)$$

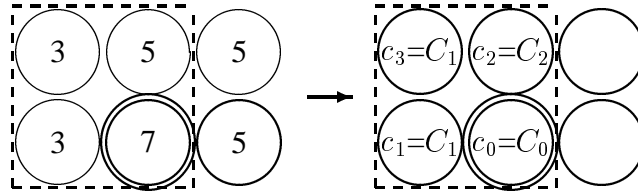
4.5.5 Exemple

Voyons maintenant avec un exemple comment ce modèle se comporte. Dans cet exemple on a $N = 7$, $N' = 2$ et $|\mathcal{N}_i| = 3$. Dans les figures suivantes le point courant est identifié par un cercle double, et le voisinage du point courant est inscrit dans un carré pointillé. Ces figures illustrent la tranformation utilisée pour le modèle à paramètres réduits.

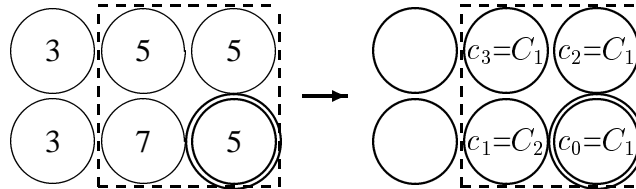
Dans les exemples de la figure 4.2, on voit que la même valeur d'étiquette peut-être transformée de manière différente selon le contexte : l'étiquette au centre de la rangée supérieure de l'exemple 1 et 2 se transforme respectivement en symbole C_2 et C_1 . L'exemple 1 démontre aussi un cas où le symbole courant ne figure pas parmi les N' symboles les plus fréquents de son voisinage, ce qui requiert un code supplémentaire, ou l'usage du terme de normalisation $1/|T_i^{-1}(c_0)|$ au sens de l'équation (4.58).

On voit aussi que l'exemple 2 et 3, dont les valeurs d'étiquettes sont tous différentes, se transforment en un même contexte. ce qui nous permet de réutiliser les mêmes paramètres du modèle pour exprimer la probabilité de ces deux contextes. Ce dernier point illustre bien

Exemple 1 : $v_i = (3, 5)$



Exemple 2 : $v_i = (5, 7)$



Exemple 3 : $v_i = (4, 3)$

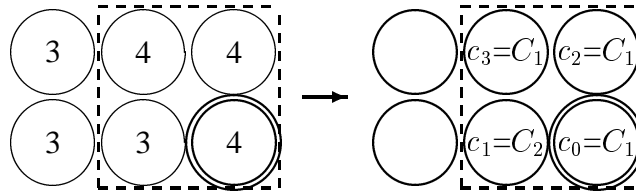

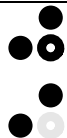







FIG. 4.2 – Transformation des valeurs d'étiquettes

l'utilité de notre représentation. Le modèle exprime la probabilité d'une manière à préserver la structure spatiale ainsi que l'information de similarité des étiquettes du voisinage dans un grand nombre de contextes.

Dans le tableau 4.1 nous présentons un modèle complet de probabilité pour un système à deux voisins ($|\mathcal{N}_i| = 2$), et une transformation à $N' = 2$ classes de similarité. C'est une illustration du genre de modèle probabiliste que nous pouvons exprimer avec le modèle à paramètres réduits. Ce modèle complet n'a que 5 paramètres.

contexte	événement	probabilité conditionnelle $\Pr(\text{événement} \mid \text{contexte})$
		très élevée
		très faible
		élevée (transition verticale)
		élevée (transition horizontale)
		faible

TAB. 4.1 – Structure spatiale des classes d'équivalence et probabilités associées

Chapitre 5

Représentation du mouvement

Nous verrons dans ce chapitre comment différentes représentations du mouvement peuvent être intégrés au problème de minimisation débit-distorsion.

5.1 Composantes de la représentation du mouvement

Voyons d'abord d'une façon générale les entités qui sont utilisées pour représenter le mouvement, et qui seront combinés et spécifiés différemment pour obtenir une représentation spécifique d'une séquence vidéo. Ces composantes sont : les régions, le mouvement propre, et l'image ou le résiduel.

5.1.1 Les régions

Une image de la séquence représentée est presque toujours divisée en régions. Dans le cas le plus simple, l'image est divisée en blocs rectangulaires de dimensions fixes et connues. Un raffinement possible est de permettre des dimensions variables en définissant une hiérarchie de blocs qui peuvent être subdivisés de manière récursive. Un autre raffinement possible consiste à faire la partition du plan d'image par tessellation d'une autre forme que le rectangle, encore une fois si la forme de base s'y prête on peut appliquer une représentation hiérarchique. De manière plus générale, on peut encoder des régions arbitraires

du plan, par contre on limite souvent ce genre de représentation aux régions définies par des contours simples (régions simples connexes). La figure 5.1 illustre quelques possibilités.

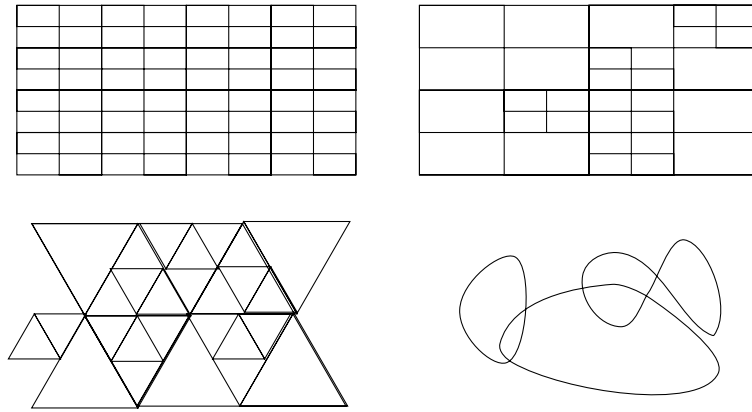


FIG. 5.1 – Différents modèles de contour : fixé, hiérarchique, hiérarchique non-rectangulaire, arbitraire

Lorsqu'on utilise les régions pour la compensation en mouvement on obtient implicitement (selon le modèle de mouvement) une description de régions dans l'image de prédiction. Le modèle peut permettre (ou pas) la superposition des régions ce qui mène à des modes de compensation différents. La figure 5.2 tente d'illustrer ce concept.

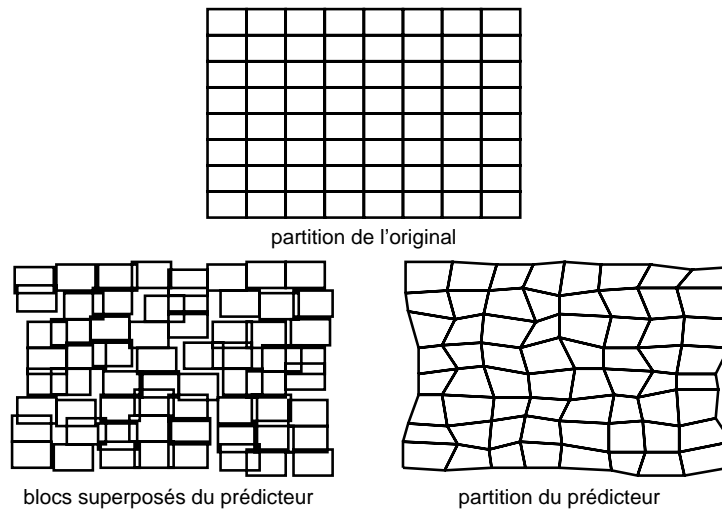


FIG. 5.2 – Régions superposées ou partition dans l'image de prédiction

Le codage

Le codage de ces structures doit être adapté. Pour les blocs de dimension fixe et connue, par exemple, aucune information ne doit être stockée ou transmise puisque toute l'information de contour est connue d'avance.

Pour la représentation hiérarchique on fixe typiquement la grandeur minimale et maximale des blocs et la structure des blocs est représentée par un arbre de division récursive. Il faut parfois s'assurer que ces contours forment une partition exacte du plan image : par exemple que les dimensions de l'image soient divisible par la taille maximale des blocs.

Pour les contours arbitraires, on a beaucoup plus de choix, mais la courbe qui définit la frontière de la région est normalement représentée par un suite de points (chain coding) ou par des courbes paramétriques spécialement conçues (spline fitting). Dans ce dernier cas il est possible que l'on puisse faire varier la fidélité d'un contour idéal par le nombre de points de contrôle des courbes, ou par la quantification des paramètres définissant ces courbes.

Pour intégrer ces représentations au modèle débit-distorsion, nous pourrions ainsi faire varier la structure des contours. C'est d'ailleurs ce qui est réalisé dans [Lee98, SSG96].

5.1.2 Le mouvement

Le mouvement sert à relier des positions spatiales de différentes trames. Cette information est souvent utilisée pour effectuer la prédiction d'une image à partir d'une autre. *e.g.* Un prédicteur $P(x, t)$ de l'image $I(x, t)$ pourrait être défini par

$$P(x, t) = I(M_i(x), t - 1) \quad \forall x \in B_i \quad (5.1)$$

où la fonction $M_i : x \mapsto x'$ représente le modèle de mouvement pour la région i (voir aussi la section 2.3) .

La figure 5.3 démontre bien le genre de déplacements que les différents modèles peuvent représenter : Translation pure, deux différentes transformations affines, et une transformation bilinéaire.

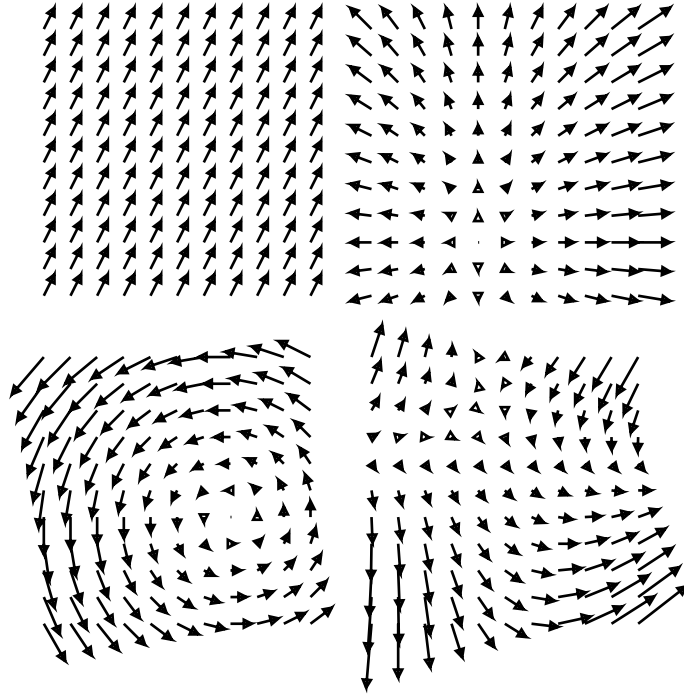


FIG. 5.3 – Exemples du mouvement pour différents modèles.

Lorsque l'image est définie sur un réseau d'échantillonnage régulier il est fréquent que la position $x' = M_i(x)$ n'appartienne pas au réseau d'échantillonnage, dans ce cas il faut faire appel à un interpolateur [Key81].

Translation Translation simple et constante pour toute une région. C'est le cas qui est utilisé dans la majorité des standards de compression. Dans ce cas on a

$$M_i(x) = x + d_i \tag{5.2}$$

où x et d_i sont des vecteurs de position dans l'image.

Transformations affines En permettant des formes plus variées de M on permet de former des prédicteurs plus fidèles. Par exemple la famille des transformations affines (plus les translations).

$$M_i(x) = A_i x + d_i \tag{5.3}$$

Ce modèle représente un champ de mouvement qui varie avec la position dans la région. Les autres modèles de la section 2.3 offrent une généralisation de ces transformations. La

représentation des paramètres libres n'est pas fixée par le modèle. Par exemple pour les transformations affines nous devons spécifier six paramètres : A_i et d_i . Nous pourrions quantifier et encoder directement ces paramètres. De manière équivalente nous pourrions spécifier la transformation par trois vecteurs de mouvement.

$$M_i(x_j) = v_{ij} \quad j = 1, 2, 3 \quad (5.4)$$

Tant que les positions x_j ne sont pas co-linéaires nous pouvons déduire les valeurs de A_i et d_i à partir des *échantillons* v_j . Le choix de la représentation sera très importante dans le contexte où l'on veut quantifier les paramètres du modèle, *e.g.* en utilisant l'approche de quantificateur à résolution variable.

Il faut parfois analyser d'autres caractéristiques du modèle. Par exemple la capacité d'avoir des régions superposées par opposition à des régions qui forment toujours une partition du plan image. Ces facteurs peuvent être critiques pour la représentation efficace de phénomènes d'occlusion.

Donc pour le mouvement, comme pour les régions, les différentes structures doivent être adaptées pour le codage. Le support ou la résolution des informations peut être variable mais elle est typiquement identifiée aux régions : par exemple un vecteur par bloc ou encore une transformée affine et une translation par région.

5.1.3 L'image ou le résiduel

Dans le cas où l'on utilise une compensation en mouvement on encode typiquement le résiduel de la prédiction par un codage par transformée. Donc le codage est fait à deux niveaux, une boucle DPCM dans la direction du mouvement et une transformation du résiduel par la DCT par exemple. De plus, ce résiduel est quantifié de manière variable. Typiquement, un paramètre de quantification est aussi associé à chaque région. Dans le cas où les régions sont de grandeur et potentiellement de forme variable, il faut adapter la transformée au support de la région. Lorsque le mouvement est modélisé par des trajectoires de mouvement, cette adaptation à la forme du contour doit aussi s'appliquer à la direction temporelle

(ou à la direction du mouvement).

Le codage

Ici aussi le choix des paramètres de quantification peut être relié à notre modèle débit-distorsion.

5.2 Quantification en amplitude du mouvement

Dans ce deuxième problème nous désirons encoder une image par la méthode suivante :

- On suppose qu’une image précédente a été transmise, et nous avons accès à une copie locale (encodeur) de l’image précédente reconstruite. De plus on suppose qu’un champ de déplacement entre l’image courante et l’image précédente reconstruite a été estimé.
- Division de l’image en blocs de taille fixe.
- Quantification des vecteurs de mouvement à une résolution Δ *e.g.* déplacement par pixel entier, demi-pixel, quart de pixel, etc.
- Compensation en mouvement de chaque bloc (obtention du résiduel de prédiction)
- Transformation par DCT du résiduel de prédiction pour chaque bloc.
- Quantification du bloc i par un facteur de quantification q qui est connu et fixé pour tous les blocs de l’image. La quantification est similaire au problème précédent sauf que les facteurs de quantification associés aux différents coefficients DCT sont différents.
- Les coefficients de la DCT quantifiés sont ensuite codés avec un codage entropique par plages.

Soient $I(x, t)$ une séquence d’images et $\hat{I}(x, t)$ la séquence reconstruite après décodage. Soient les vecteurs de mouvement $d_i = (d_{x_i}, d_{y_i})$ associés au bloc i . Soient les pas de quantification $\Delta = (\Delta_x, \Delta_y)$ pour la quantification en amplitude de vecteurs de mouvement associés à chaque bloc (Δ_x pour la composante horizontale et Δ_y pour la direction

verticale).

Le résiduel de prédiction est défini par

$$DPD(x, t, \Delta) = I(x, t) - \hat{I}(x + d_i^{\Delta_i}, t - 1) \quad \forall x \in B_i$$

où la compensation en mouvement est effectuée par une version quantifiée du champ de vecteurs $d_i, d_i^{\Delta_i}$.

On a alors

$$D_i(\Delta_i) = \sum_{x \in B_i} [I(x, t) - \text{DCT}_{B_i}^{-1}(Q(\text{DCT}_{B_i}(DPD(x, t, \Delta_i)), q)) - \hat{I}(x + d_i, t - 1)]^2$$

et

$$R_i(\Delta_i) = R^{(d)}(d_i^{\Delta_i}) + R^{(DPD)}(d_i^{\Delta_i})$$

où $R^{(d)}$ et $R^{(DPD)}$ sont respectivement les coûts de codage des vecteurs de mouvement, et du résiduel.

Comme dans le cas des facteurs de quantification nous avons essayé deux versions de ce problème. La première où la quantification des vecteurs de mouvement était constante pour toute l'image, et une deuxième où la quantification des vecteurs de mouvement était variable pour chaque bloc. Les résultats montrent aucun gain à l'adaptation de la quantification des vecteurs de mouvement à chaque bloc.

5.2.1 Solution du problème

De manière analogue au problème simplifié (3.3) nous produisons l'enveloppe convexe des points réalisables, i.e. associés aux différentes combinaisons de paramètres q et Δ . (*Ce graphique n'est pas assez complet pour démontrer les résultats adéquatement.*) Sachant que les courbes qui semblent se distinguer sont associées aux différents Δ des vecteurs de mouvement (1 pixel, 1/2, 1/4,...), on peut voir dans la figure 5.4 que la résolution optimale des vecteurs de mouvement varie avec λ . À plus haut débit, la résolution optimale des vecteurs de déplacement est de 1/8 ou 1/16 de pixel, tandis qu'à faible débit elle est plutôt de 1 ou 1/2 pixel.

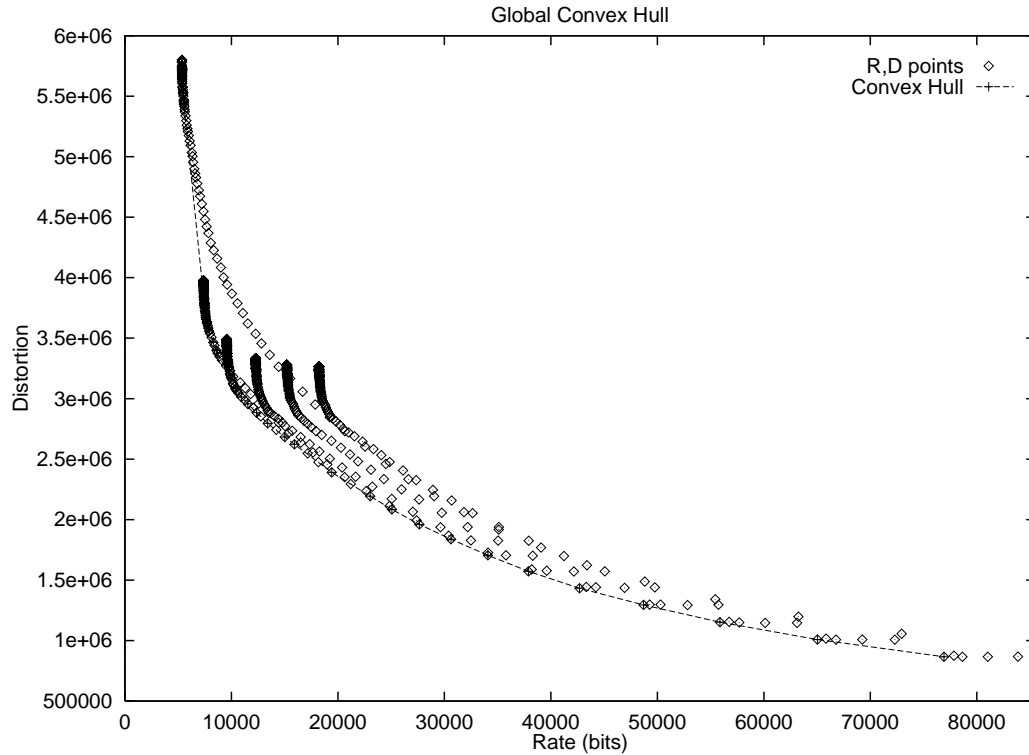


FIG. 5.4 – Enveloppe convexe des points R-D réalisables avec différentes combinaisons q et Δ

Dans la figure 5.5 on voit des courbes de débit-distorsion optimales où chaque point sur une courbe représente un couple R, D optimal (et a donc un facteur de quantification global q et un pas de quantification global pour les vecteurs de mouvement Δ optimaux associés).

Dans ce modèle la prédiction est basée sur une image précédemment reconstruite, donc les différentes courbes représentent la performance optimale du codec pour un facteur de quantification q_{Pred} de l'image précédente. On voit que plus l'image reconstruite à été dégradée par la quantification moins le codeur est efficace.

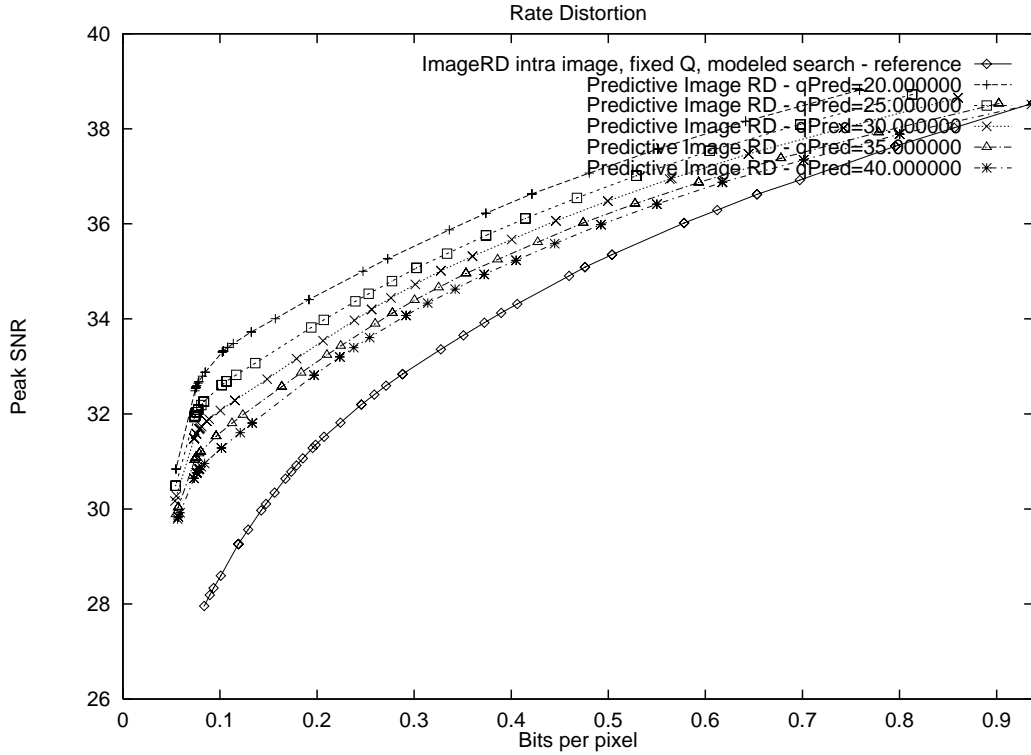


FIG. 5.5 – Courbes débit-distorsion pour la quantification en amplitude du champ de déplacement

5.3 Représentation du mouvement par un dictionnaire de modèles

Dans cette section nous décrivons une représentation du mouvement basée sur un dictionnaire de modèles de mouvement ainsi que les méthodes d'estimation associées. Les champs de mouvement sont représentés par des champs polynômiaux de faible degré et une carte d'étiquettes associée. Nous employons une méthode de codage entropique contextuelle novatrice et adaptée a cette représentation.

Étant donné que la flexibilité des paramètres d'encodage nous ouvrent des portes pour l'optimisation débit-distorsion subséquente, nous nous sommes permis d'explorer une représentation qui dépasse le cadre classique de la représentation du mouvement basé sur la translation simple de blocs.

Le mouvement apparent de séquences vidéo est typiquement constitué de quelque modèles simples. Pour exploité cette observation nous représentons ces quelque modèles simples dans un dictionnaire de modèles et nous indiquons par une étiquette quel modèle utiliser à chaque position spatiale dans l'image. Tel que mentionné plus haut les modèles sont composés de champs polynômiaux de faible degré. Cette représentation nous permet d'associer chacun des modèles de mouvement à une région arbitraire (et possiblement non-connexe) de l'image sans recours explicite à des techniques de segmentation ou de représentation des contours.

Nous présentons aussi un méthode efficace pour l'estimation et le codage de cette nouvelle représentation. Le processus itératif d'estimation de la représentation raffine successivement le dictionnaire de modèles et la carte d'étiquettes.

Tel que mentionné dans la section 5.1.2 il est possible de faire la compensation en mouvement par un modèle plus élaboré que des simples translations. Nous voulons inclure ces modèles dans notre représentation. e.g. :

$$M(p) = \begin{bmatrix} u_x^0 \\ u_y^0 \end{bmatrix} \text{ ou } M(x, y) = \begin{bmatrix} u_x^0 + u_x^1 x + u_x^2 y \\ u_y^0 + u_y^1 x + u_y^2 y \end{bmatrix} \text{ ou } M(x, y) = \begin{bmatrix} u_x^0 + u_x^1 x + u_x^2 y + u_x^3 xy \\ u_y^0 + u_y^1 x + u_y^2 y + u_y^3 xy \end{bmatrix}$$

De manière générale :

$$M(p) = \begin{bmatrix} \sum_m u_x^m g_m(p) \\ \sum_m u_y^m g_m(p) \end{bmatrix},$$

où les $g_m(p)$ sont des monômes en fonction de la position spatiale $p = (x, y)$. Un modèle de mouvement est alors complètement déterminé par les fonctions de bases g_m et les constantes u_x^m, u_y^m . Nous définissons le degré du modèle comme étant le degré maximum des polynômes g_m , par exemple la transformée bilinéaire ci-haut est de degré 2. Un modèle représente une valeur de déplacement à n'importe quel endroit dans l'image. En utilisant ces notions nous proposons la représentation suivante pour le champs de mouvement.

5.3.1 Représentation

Soit un dictionnaire D composé de N modèles de mouvement M_i tels que décrits plus haut,

$$D = \{M_1, \dots, M_N\},$$

et une carte d'étiquettes $T(x) \in \{1, \dots, N\}$ qui associe un modèle de mouvement à chaque position x de l'image.

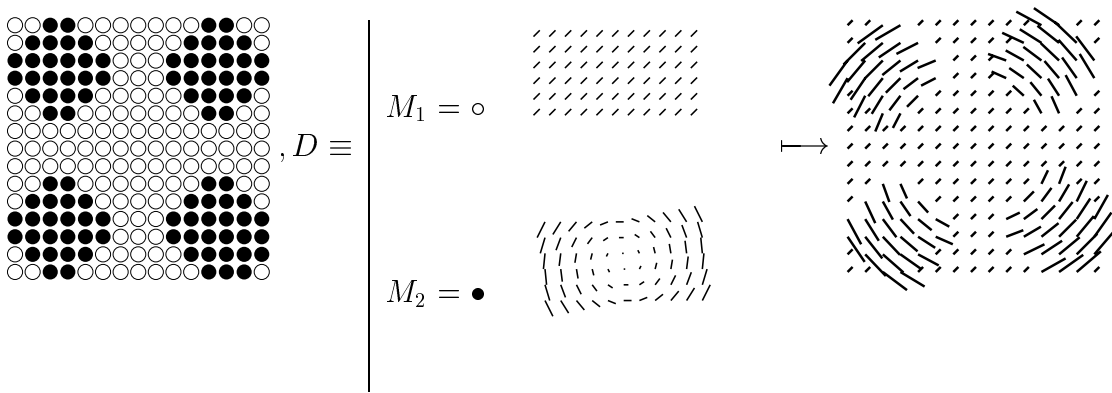


FIG. 5.6 – Carte d'étiquettes, dictionnaire et champs dense reconstruit

La figure 5.6 illustre les composantes de notre représentation, soit le dictionnaire D et la carte d'étiquette T . On voit aussi comment ces deux composantes sont combinées pour former un champ de mouvement dense sur l'image entière.

$$M(x) = M_{T(x)}(x) \quad \forall x \in \text{Image}.$$

Par un raffinement de la représentation, nous pouvons fixer explicitement la valeur de l'étiquette sur une petite région, comme un bloc de 4x4 ou 8x8 pixels, ce qui nous permet de représenter la carte d'étiquettes à plus faible résolution. Notons cependant que même si la valeur de l'étiquette est fixée dans ce petit bloc le mouvement reconstruit $M(x)$ peut encore varier selon le modèle.

5.3.2 Estimation

Le processus d'estimation de la représentation produit simultanément le dictionnaire et la carte d'étiquettes associée. Nous pouvons procéder de plusieurs façons. Nous pouvons estimer la représentation qui approxime un champs indépendamment estimé ou connu, ou encore nous pouvons trouver la représentation qui minimise directement le résiduel de prédiction induit par la représentation.

$$\min_{D,T} \|M(D, T) - M_{\text{connu}}\|,$$
$$\min_{D,T} \|I - \hat{I}_{D,T}\|.$$

Nous pouvons concevoir notre représentation du mouvement comme une forme de quantification vectorielle où le dictionnaire définit un certain nombre de niveaux représentatifs pour les champs de mouvement globaux. De cette analogie nous pouvons définir le processus d'estimation comme un algorithme généralisé de Lloyd (GLA) [GG91] pour l'optimisation d'un quantificateur.

Comme ça le dictionnaire définit un ensemble de champs de mouvement globaux et l'étiquette associée a chaque position identifie lequel de ces modèles a été choisi.

Pour définir le dictionnaire (quantificateur) nous devons spécifier un nombre de modèles représentatifs, et aussi comment chaque valeur admissible d'un vecteur de mouvement sera associé à un de ces modèles. Dans l'algorithme de Lloyd (GLA) le quantificateur optimal est obtenu par raffinement successif par l'application tour à tour des deux règles suivantes.

Répartition au voisin le plus proche Cette étape attribue des valeurs aux étiquettes étant donné un dictionnaire de modèles.

Calcul du centroïde Cette étape ré-estime les modèles de mouvement étant donné un carte d'étiquettes.

5.3.3 Affectation d'étiquettes

Dans le langage de l'algorithme de Lloyd, cette étape est la répartition au voisin le plus proche.

Dans ce cas, ceci revient à trouver l'étiquette à chaque position qui est associée au modèle du dictionnaire qui minimise le critère de distorsion locale pour le vecteur à cette même position.

$$T(p) = \min_{t \in \{1, \dots, N\}} d(M_t(p)) \quad (5.5)$$

où $d(M_t(p))$ est la mesure de distorsion pour le $t^{\text{ième}}$ modèle à la position p . E.g. nous pouvons utiliser

$$d(M_t(p)) = \|M_t(p) - M_{\text{connu}}(p)\| \quad (5.6)$$

pour un champs de mouvement connu, ou encore

$$d(M_t(p)) = \|I(p) - \hat{I}(p - M_t(p))\| \quad (5.7)$$

pour l'image courante I et la trame précédente reconstruite \hat{I} ce qui revient à minimiser l'erreur de prédiction compensée mouvement directement.

5.3.4 Ré-estimation des modèles

Dans le langage de l'algorithme de Lloyd, cette étape est le calcul du centroïde

Dans notre cas, ceci revient à réestimer les paramètres du modèle de mouvement pour chaque modèle dans le dictionnaire. Ce calcul de réestimation étant basé sur l'ensemble des positions qui ont été affectés à ce modèle par la carte d'étiquettes.

$$M_i \equiv \min_u \sum_{\{p|T(p)=i\}} d(M_i(p)) \quad \forall i \quad (5.8)$$

Cette minimisation est effectuée par une méthode de moindres carrés pour les paramètres u et une mesure de distorsion appropriée pour M_i .

Il est aussi possible de trouver par un algorithme de descente dans les paramètre u . Cet algorithme ré-estime directement chacun des modèles de mouvement. Ce processus itératif produit une séquence convergente de modèles M_i^j , où on fixe M_i^0 au modèle de mouvement précédemment utilisé et $M_i^j(p) \equiv M_i^{u_j}(p)$ avec

$$u_{j+1} = u_j - \nabla_2 \sum_{\{p|T(p)=i\}} d(M_i^{u_j}(p))^{-1} \nabla \sum_{\{p|T(p)=i\}} d(M_i^{u_j}(p)) \quad (5.9)$$

5.4 Codage de la carte d'étiquettes

Dans le contexte de la représentation du mouvement par dictionnaire pour stockage ou transmission, certaines informations doivent être reproduites au décodeur.

- résolution de l'image
- résolution de la carte étiquette
- nombre de modèles dans le dictionnaire
- le degré et les paramètres de chacun des modèles de mouvement.
- la carte étiquette elle-même.

Toutes ces informations sauf la carte étiquette sont codées par des moyens simples parce qu'ils ne représentent qu'une très petite partie du taux total. Dans notre schème de codage même les paramètres des modèles de mouvement sont codés à pleine résolution des nombres réels des systèmes informatiques utilisés et ne subissent pas d'autre transformation ou quantification. En pratique, si le nombre de modèles devient grand il faudra se pencher sur un codage efficace de ces paramètres.

L'effort de codage est donc concentré sur la représentation de la carte étiquette. L'étiquette pour chaque position de l'image est codée par code arithmétique conditionnel.

Le conditionnement est basé sur un voisinage causal de chaque étiquette. L'utilisation d'un code arithmétique optimisé nous permet de coder chaque étiquette avec environ $-\log_2(\Pr[T(\mathbf{p})])$ bits. L'optimalité du code est donc fortement dépendant de la qualité du modèle probabiliste de la carte étiquette $\Pr[T(\mathbf{p})]$.

Une des propriétés de la carte étiquette que nous souhaitons refléter dans le modèle est que les différentes valeurs des étiquettes soient considérées comme discrètes et non reliée par une mesure de distance. Même si ces étiquettes sont identifiées numériquement, la valeur étiquette 4 n'est nécessairement plus *proche* de la valeur 5 que ne l'est une valeur de 1. La seule mesure de distance appropriée en serait une qui discrimine *l'égalité* ou *l'inégalité* de deux étiquettes.

Nous désirons cependant exploiter une certaine partie de la structure de voisinage spatial, qui se produit avec des probabilités hautement asymétriques nous rappelant que c'est

cette asymétrie dans la probabilité d'apparition que nous exploitons finalement par le codage entropique.

Nous utilisons donc un modèle de codage conditionnel défini au chapitre 4.

5.5 Codage entropique conditionnel contraint sur le taux

Le processus qui détermine un quantificateur peut être modifié pour tenir compte du coût de codage des symboles.

Il est possible de déterminer un quantificateur optimal conjointement avec le code entropique qui servira à encoder les symboles quantifiés. Ce processus se nomme *Entropy constrained quantization* [CL90], et un raffinement de cette méthode nous permet même de déterminer un quantificateur et un codage conditionnel (du type qu'on utilise) qui soient aussi conjointement optimaux.

Comme nous avons vu dans la section 5.4, nous pouvons calculer le coût de codage associé aux étiquettes à chaque position. Dans cette approche il s'agit de déterminer à chaque itération, un code optimal pour la carte d'étiquettes et la règle du voisin le plus proche est modifiée pour inclure un biais en fonction du coût de codage de chaque étiquette.

$$d'(M_t(p)) = d(M_t(p)) + \lambda R(T(p)) \quad \lambda > 0. \quad (5.10)$$

L'algorithme d'estimation du dictionnaire de modèles devient donc

Répartition au voisin le plus proche Cette étape attribue des valeurs aux étiquettes étant donné un dictionnaire de modèles, et utilise la distance modifiée présentée ici.

Calcul du centroïde Comme avant, cette étape ré-estime les modèles de mouvement étant donné un carte d'étiquettes.

Optimisation du code entropique Cette étape produit un nouveau code entropique pour le codage d'étiquettes tel que défini au chapitre 4.

Chapitre 6

Simulations et analyse

Nous présentons dans cette section les résultats de plusieurs expériences de simulations informatiques qui ont été réalisées pour illustrer les idées présentées dans les chapitres précédents.

6.1 Quantification en amplitude des vecteurs de mouvement

Présenter un tableau/graphes pour la simulation qui détermine q et Δ fixe, et variable pour une image, avec predicteur de différente qualité

6.2 Codage de cartes d'étiquettes

Nous avons décrit au chapitre 4 une méthode de codage pour une séquence d'étiquettes avec différents modèles. Nous regroupons ici des simulations de codage de telles séquences. Premièrement nous illustrons le codage d'étiquettes avec les modèles sans mémoire et avec le modèle de premier ordre. Les sources sont pseudo-aleatoires avec une variété de propriétés probabilistes. Les résultats du codage sont illustrés dans le tableau 6.1.

Test	N, M	optimal		paramètres réduits		
		sans mémoire	ordre 1	sans marginale	approx.	exacte
1	3, 30	1.549398	0.996529	1.01711	1.021037	1.011047
2	5, 100000	2.321908	2.321756	2.321926	2.321921	2.321921
3	2, 100000	0.999997	0.4684	0.468425	0.468425	0.468425
6	3, 100000	1.584951	1.365108	1.365174	1.365168	1.365167
9	4, 100000	1.999991	1.89583	1.895905	1.895889	1.895887
12	5, 100000	2.321904	2.269149	2.269264	2.269219	2.269216
15	2, 100000	0.878548	0.50842	0.526046	0.526046	0.526046
21	2, 100000	0.990376	0.813258	0.816545	0.816545	0.816545
22	2, 100000	0.295503	0.057905	0.0675	0.0675	0.0675
25	3, 100000	1.248225	0.821093	0.931512	0.911371	0.903442
32	3, 100000	1.531732	1.066705	1.099717	1.083186	1.079822
33	3, 100000	0.529566	0.204837	0.255149	0.253449	0.253099
36	4, 100000	1.815223	1.402877	1.505899	1.424586	1.421652
37	4, 100000	1.761621	1.123078	1.216364	1.148436	1.141639
39	4, 100000	1.980262	1.24689	1.378855	1.352253	1.350528
45	5, 100000	2.27443	1.716382	1.810896	1.793314	1.792129
46	5, 100000	2.236981	1.523863	1.633113	1.626018	1.625394
54	5, 100000	2.271837	1.633771	1.750657	1.740036	1.73943

TAB. 6.1 – Codage d'étiquettes avec modèles d'ordre 0 et 1

Ensuite nous présentons les résultats de simulations de codage d'étiquettes avec le modèle à voisinage élargi pour une collection d'image étiquettes représentatives. Les images étiquettes sont décrites dans la figure 6.1.

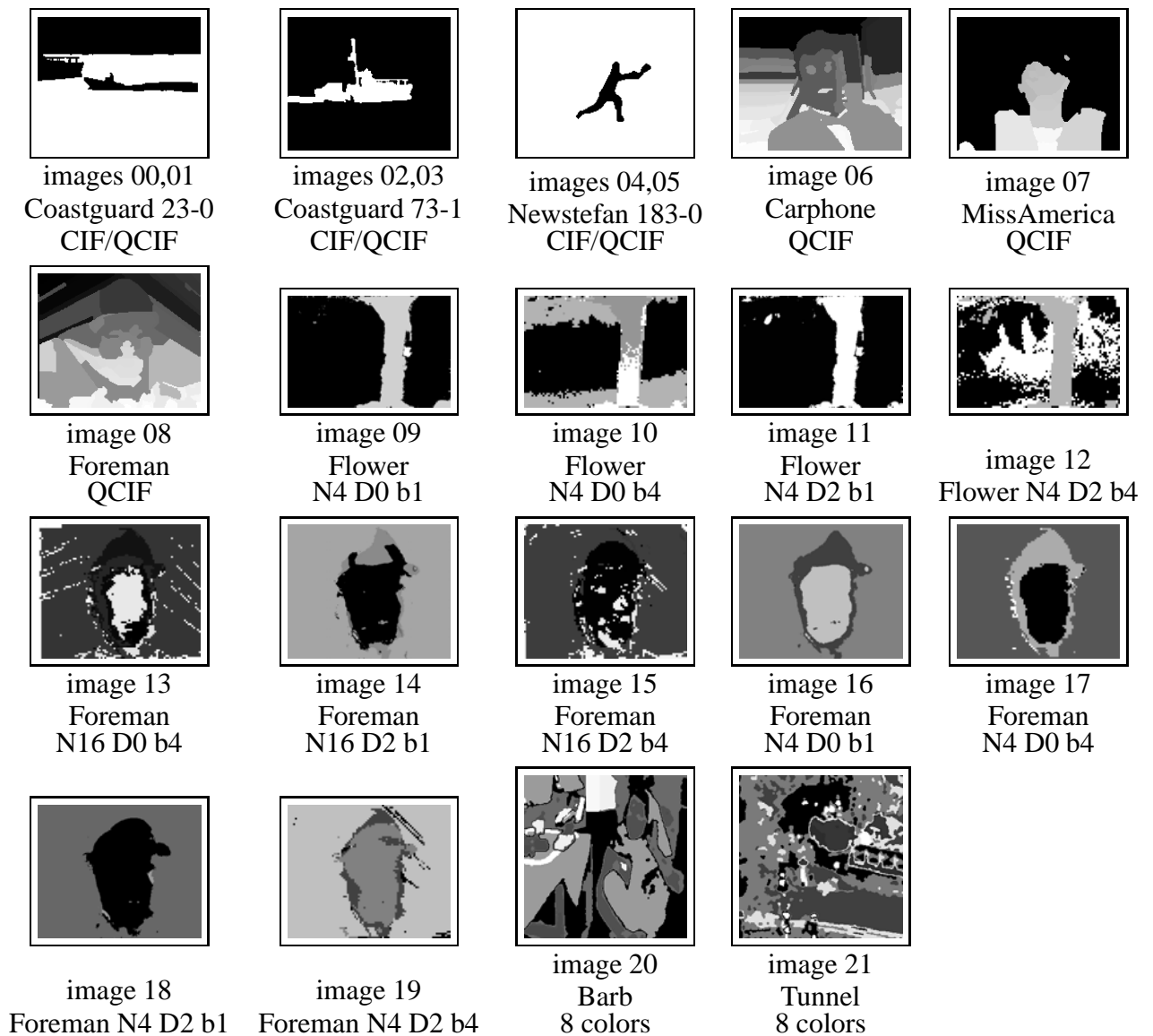


FIG. 6.1 – Cartes étiquettes représentatives

Les images 00 – 05 sont des masques binaires des séquences test MPEG-4. Les images 06 – 08 sont des images tests connues segmentées. Les images 09 – 19 ont été estimées par notre représentation du mouvement par un dictionnaire. Les images 20 et 21 sont une

représentation de cartes de chrominance.

Le résultat du codage de ces étiquettes est présenté dans le tableau 6.2. Le codage contextuel simplifié a été utilisé avec différentes tailles de voisinage, et avec ou sans l'utilisation de probabilités marginales.

image	Taille du voisinage									
	2		3		4		5		6	
	-h	+h	-h	+h	-h	+h	-h	+h	-h	+h
image 00	1.0	.9	.8	.7	.6	.5	.4	.3	.2	.1
image 01	1.0	.9	.8	.7	.6	.5	.4	.3	.2	.1
⋮										
image 21	1.0	.9	.8	.7	.6	.5	.4	.3	.2	.1

TAB. 6.2 – Coût de codage des cartes (bits) par le codage contextuel simplifié

Finalemment nous comparons le codage étiquettes sur nos images test avec d'autres méthodes connues de codage d'étiquettes, ou de codage d'image sans perte. Ces résultats se trouvent dans le tableau 6.3.

image	CBC	ACAE	EQCG	EQCB	EJBG	EJGB
00	1992	1774	2400	2400	2152	2152
01	1304	987	1136	1136	1560	1560
02	2136	2107	1936	1936	2464	2464
03	1360	1166	864	864	1728	1728
04	1296	1682	1480	1480	1936	1936
05	792	1110	632	632	1424	1424
06	5816	6902	10448	11136	14768	15544
07	2368	2605	2952	3096	6344	6448
08	7448	8828	13008	14376	15984	17648
09	6496	4747	17232	15072	5160	4952
10	4192	2412	3568	2584	4128	3376
11	9792	6584	8496	7064	8968	7728
12	8664	3983	4456	7232	4752	6776
13	9120	4838	9192	6968	10088	8472
14	8736	7310	11288	10808	12704	12304
15	7968	4137	5840	7984	8016	9496
16	5896	5128	6368	7024	7160	7824
17	2320	1555	1608	1528	3040	3024
18	4848	4211	5992	4304	6824	4928
19	4360	2449	2368	2872	3680	4120
20	17136	14692	15760	18440	16528	18936
21	21960	18318	23168	24768	23440	25416

TAB. 6.3 – Coût de codage des cartes (bits)

6.3 Représentation du mouvement à résolution variable

6.4 Représentation du mouvement par un dictionnaire de modèles

Les figures 6.2 à 6.13 démontrent le principe pour différents ordres du modèle et différents nombres d'entrées dans le dictionnaire. L'ordre du modèle spécifie l'exposant maximal dans les expressions de monômes $g_m(p) = p_x^i p_y^j$ ainsi le modèle d'ordre 1 comprend les monômes :

$$g_0(p) = 1, \quad g_1(p) = p_x, \quad g_2(p) = p_y, \quad g_3(p) = p_x p_y$$

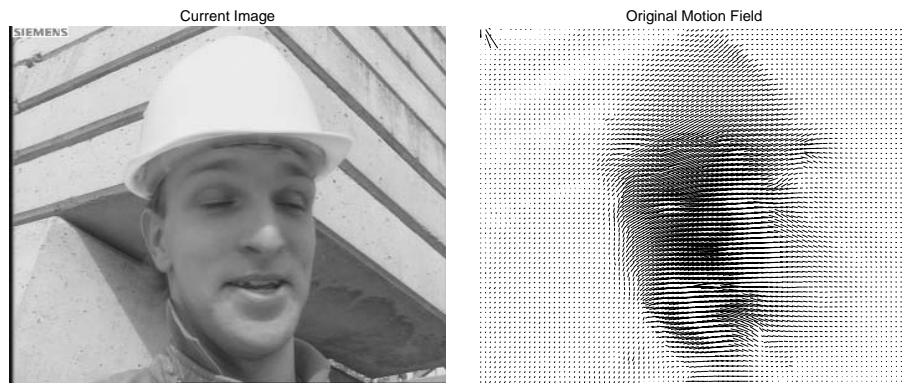


FIG. 6.2 – Champs et Image originaux

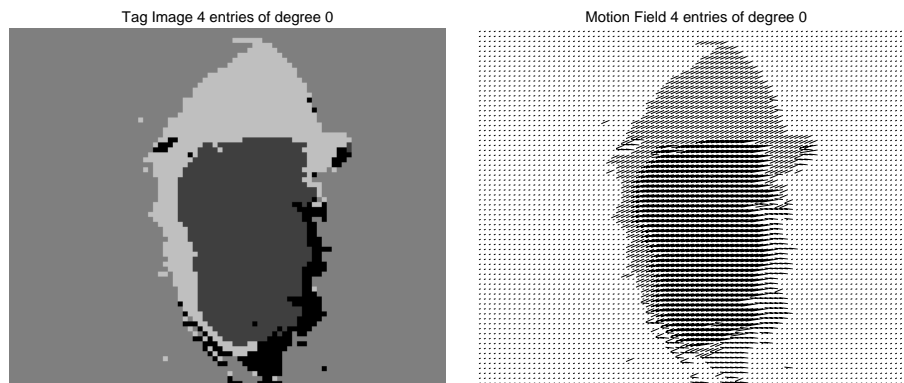


FIG. 6.3 – Champs reconstruit et carte d'étiquettes $N=4$ degré 0

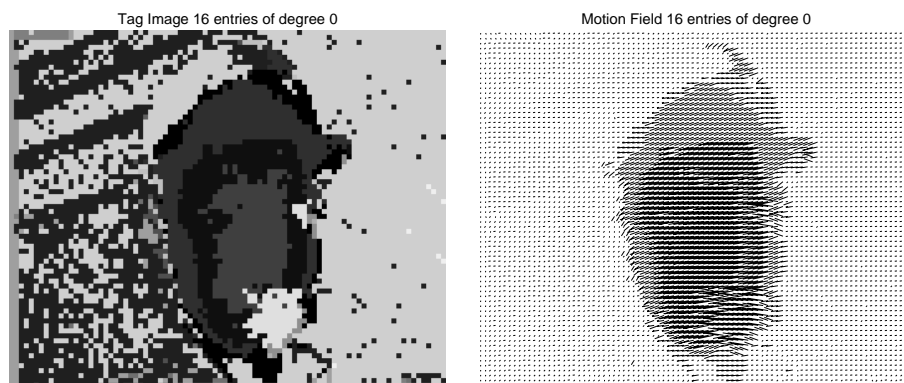


FIG. 6.4 – Champs reconstruit et carte d'étiquettes $N=16$ degré 0

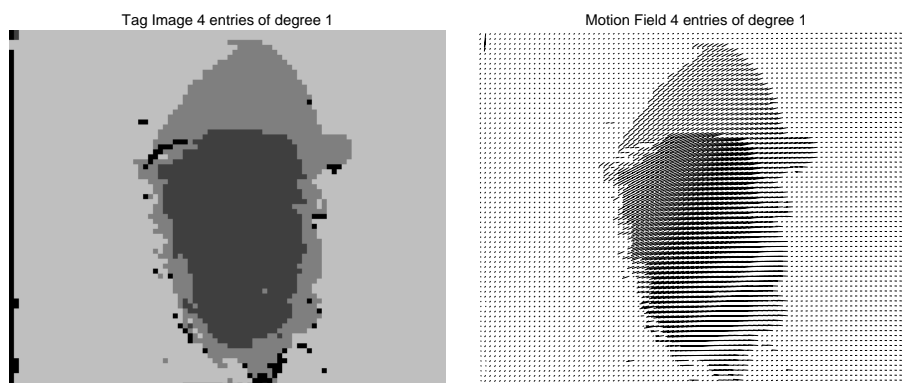


FIG. 6.5 – Champs reconstruit et carte d'étiquettes $N=4$ degré 1

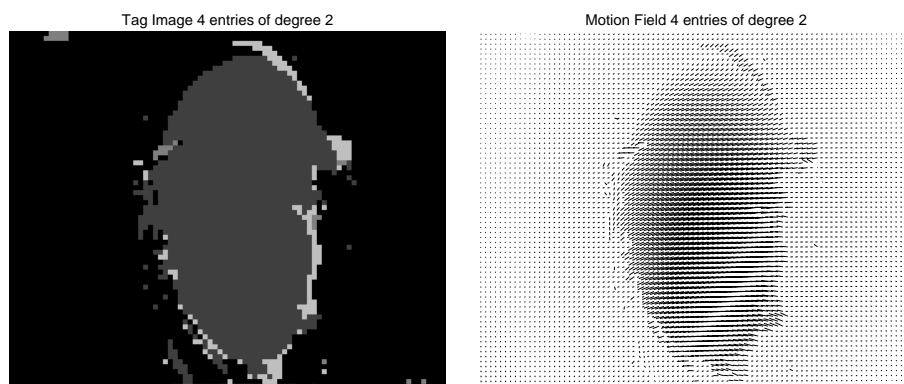


FIG. 6.6 – Champs reconstruit et carte d'étiquettes $N=4$ degré 2

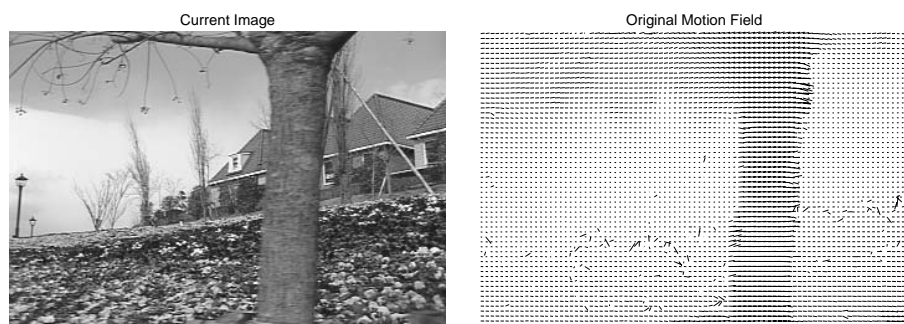


FIG. 6.7 – Champs et Image originaux

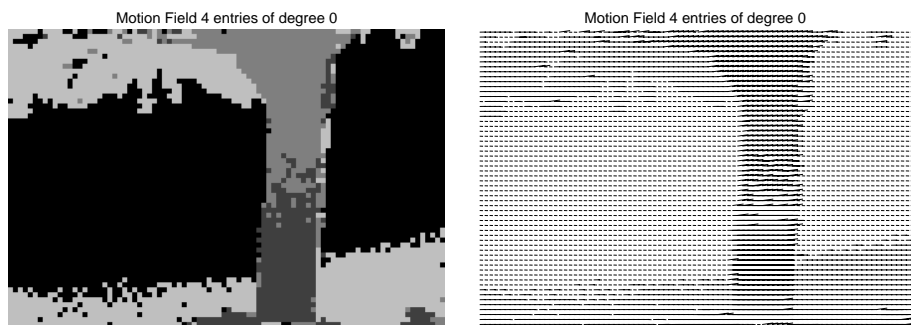


FIG. 6.8 – Champs reconstruit et carte d'étiquettes $N=4$ degré 0

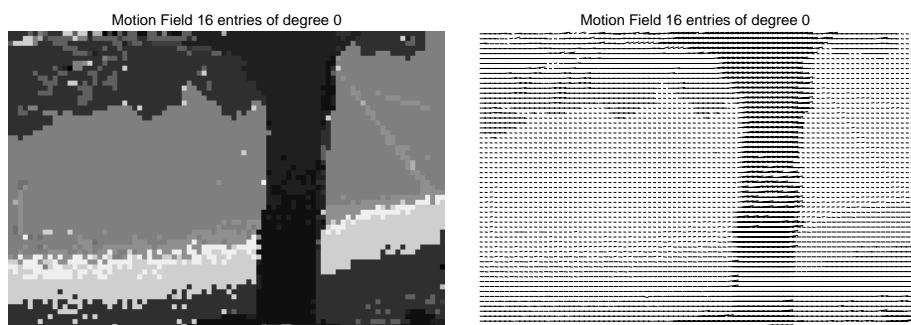


FIG. 6.9 – Champs reconstruit et carte d'étiquettes $N=16$ degré 0

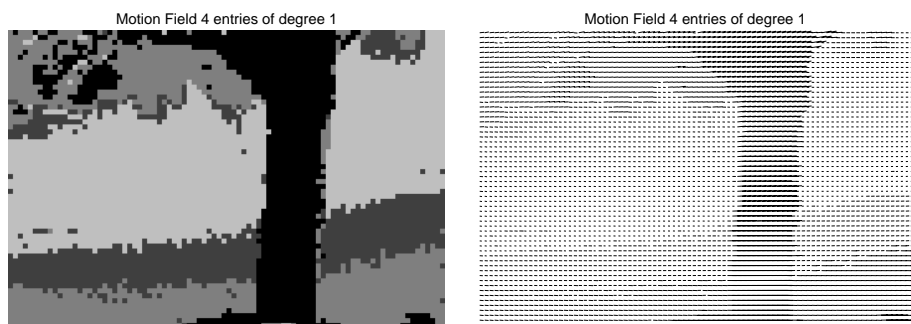


FIG. 6.10 – Champs reconstruit et carte d'étiquettes $N=4$ degré 1

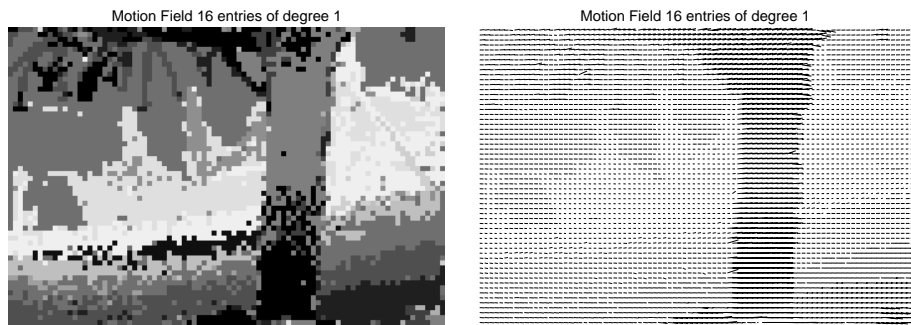


FIG. 6.11 – Champs reconstruit et carte d'étiquettes $N=16$ degré 1

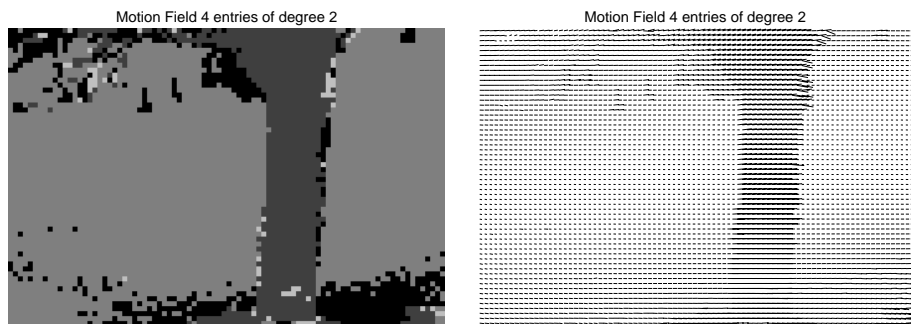


FIG. 6.12 – Champs reconstruit et carte d'étiquettes $N=4$ degré 2

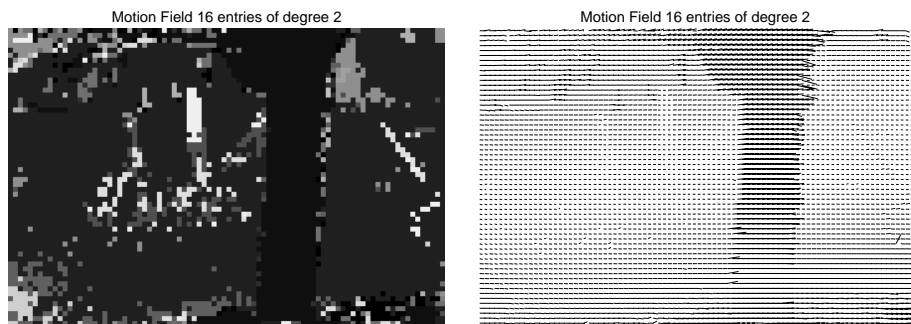


FIG. 6.13 – Champs reconstruit et carte d'étiquettes $N=16$ degré 2

6.5 Intégration dans un codec optimal débit-distorsion

6.5.1 Compensation à partir de prédicteurs de haute qualité

Nous avons réalisé un certain nombre de simulations pour vérifier la qualité de la compensation en mouvement que la représentation par dictionnaire permettait. La figure 6.14, présente des résultats utilisant deux mesures pour quantifier la performance de l'encodeur : Le coût total du codage de l'information mouvement (mesuré en bits par pixel d'image originale, et le rapport PSPR (*peak signal to prediction error ratio*). Les résultats sont comparés à la technique de compensation en mouvement de MPEG-2 à résolution plein-pixel et demi-pixel. La méthode MPEG-2 a aussi été étendue pour simuler une précision allant jusqu'à $\frac{1}{32}$ pixel de précision (ce que le standard ne permet pas). La compensation est effectuée à partir d'images haute qualité soit l'image précédente originale (non-codée, non-quantifiée).

Ainsi nous pouvons observer qu'en fixant le taux de codage, nous obtenons un gain net dans la qualité de prédiction dans tous les cas. L'amélioration pour la séquence *flower* atteint 4dB à faible taux de codage de l'information mouvement. Nous pouvons aussi observer l'impact de la nature du contenu. La séquence *flower* contient surtout de l'information de mouvement lisse associée au mouvement de la caméra et nos modèles de mouvement peuvent facilement représenter ce genre de mouvement même à de faible taux. Dans le cas de la séquence *foreman*, le mouvement est moins uniforme et les frontières de régions en mouvement sont plus complexes ce qui donne un gain en performance beaucoup plus petit.

Les résultats de la figure 6.14 incluent les meilleures combinaisons des paramètres $N \in \{4, 16\}$, $O \in \{1, 2\}$ and $b \in \{4, 8, 16\}$. La figure 6.14 démontre bien la flexibilité de notre représentation où il serait possible par une sélection judicieuse de paramètres N , O and b d'obtenir un taux de codage pour l'information mouvement ou de la qualité de la prédiction sur une très large gamme : taux allant de .005 à 0.05 bits/pixel image donnant des valeurs PSPR entre 26.5 and 29 dB pour la séquence *flower*.

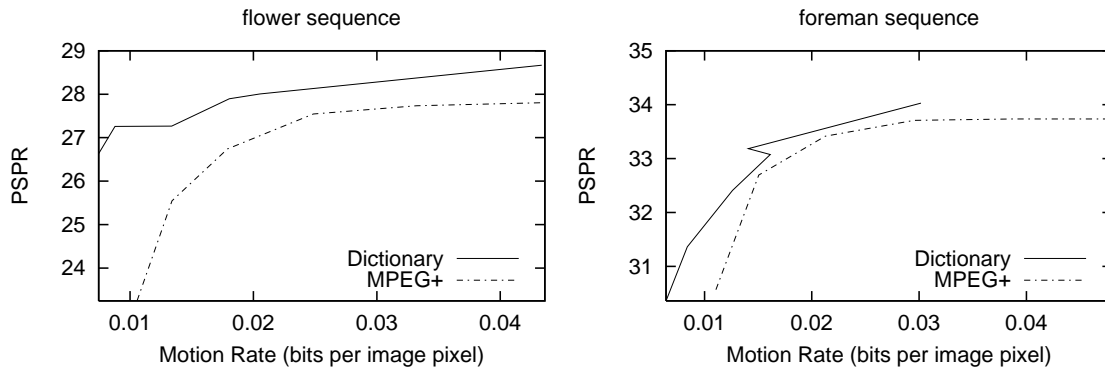


FIG. 6.14 – Qualité de la compensation en mouvement pour Flower and Foreman

La variation de la taille du voisinage pour le codage contextuel de la carte étiquette de 1 à 6 peut aussi faire varier le taux de codage de l’information mouvement d’environ 30 pourcent. La taille optimale du voisinage est typiquement de 2 à 4.

6.5.2 Simulation a distorsion constante

Nous décrivons ici des résultats tenant compte d’une implantation complète d’un cœur vidéo contraint sur le taux. Dans ces simulations, et pour obtenir un comparaison raisonnable entre le codec référence et le nôtre, les deux codecs démarrent avec la même trame reconstruite et une simple prédiction avant est effectuée pour les 30 trames suivantes. En fixant le rapport PSNR pour chaque image reconstruite, nous observons le nombre de bits nécessaires pour produire la trame reconstruite de qualité désirée. Les résultats des simulation pour les séquences *flower* (fig. 6.15) et *foreman* (fig. 6.16) sont présentés.

Les résultats sont mixtes. Pour la séquence *flower* les simulation démontrent un gain de 15 pourcent sur le taux total de codage, tandis que pour la séquence *foreman* c’est plutôt une pénalité de l’ordre de 5-10 pourcent qui est observée. La différence entre ces deux séquences est sans doute reliée a la nature du mouvement présent dans les deux séquences. La séquence *flower* peut plus facilement prendre avantage d’une information mouvement de précision supérieure.

Il est aussi intéressant de noter que l’avantage de prédiction suggérée par les simula-

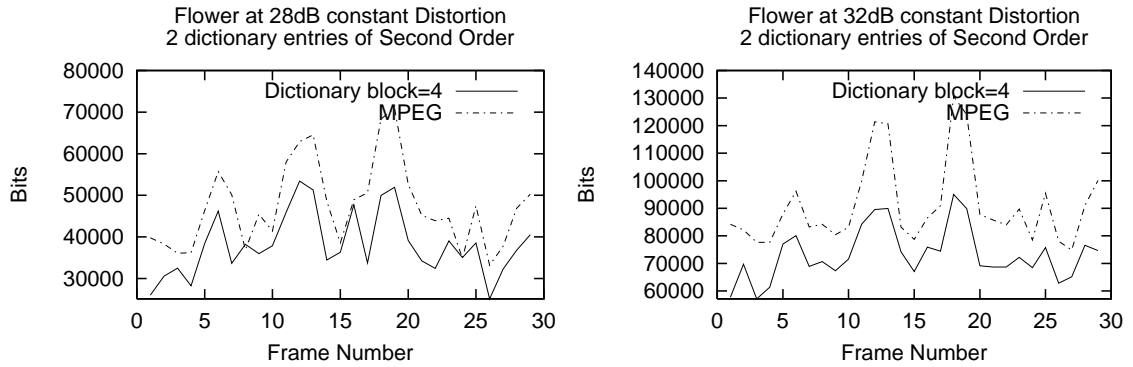


FIG. 6.15 – Résultats à distorsion constante pour la séquence Flower

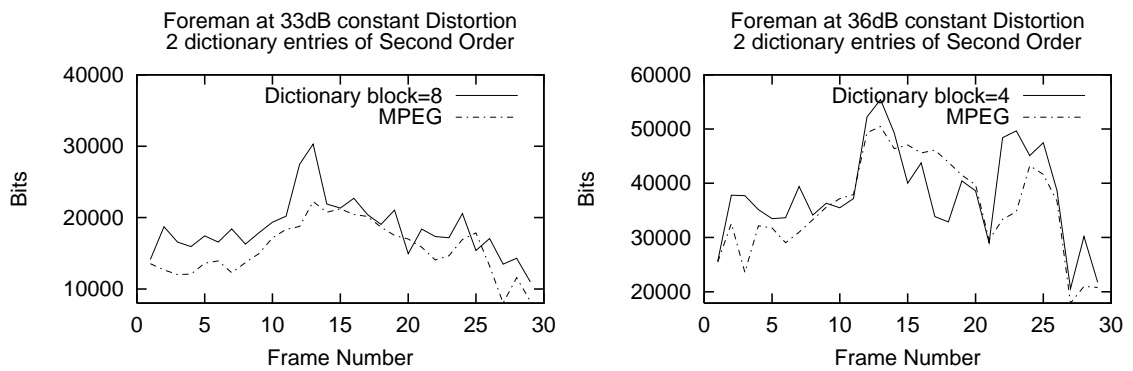


FIG. 6.16 – Résultats à distorsion constante pour la séquence Foreman

tions avec des prédicteurs de haute qualité ne c'est pas traduite par des gains aussi forts qu'espérés dans ces simulations plus réalistes. Nous croyons que c'est relié principalement a la dégradation de la qualité du prédicteur qui limite la capacité de faire une compensation en mouvement adéquate. La sensibilité à la qualité du prédicteur suggère que l'allocation inter-trame doit être examinée de plus près et qu'une compensation simple de type DPCM dans l'axe temporel est peut-être insuffisant.

Chapitre 7

Conclusions

7.1 Contributions

7.2 Travaux futurs

Bibliographie

- [BC96] Hao Bi and Wai-Yip Chan, *Rate-constrained hierarchical motion estimation using BFOS tree pruning*, Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, vol. 4, May 1996, pp. 2315–2318.
- [BD91] Claude Bergeron and Eric Dubois, *Gradient-based algorithms for block oriented MAP estimation of motion and application to motion-compensated temporal interpolation*, IEEE Trans. Circuits Syst. Video Technol. **1** (1991), no. 1, 72–85.
- [BLK95] Laurent Bonnaud, Claude Labit, and Janusz Konrad, *Interpolative coding of image sequences using temporal linking of motion-based segmentation*, Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, May 1995, pp. 2265–2268.
- [Bus97] Ralf Buschmann, *Efficiency of displacement estimation techniques*, Signal Process., Image Commun. **10** (1997), 43–61.
- [CKS95] Wilson C. Chung, Fauzi Kossentini, and Mark J. T. Smith, *Rate-distortion-constrained statistical motion estimation for video coding*, Proc. IEEE Int. Conf. Image Processing, vol. 3, October 1995, pp. 184–187.
- [CKS96] Wilson C. Chung, Fauzi Kossentini, and Mark J. T. Smith, *An efficient motion estimation technique based on a rate-distortion criterion*, Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, vol. 4, May 1996, pp. 1926–1929.

- [CL90] Philip A. Chou and Tom Lookabaugh, *Conditional entropy-constrained vector quantization of linear predictive coefficients*, Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, April 1990, pp. 197–200.
- [CLG89a] Philip A. Chou, Tom Lookabaugh, and Robert M. Gray, *Entropy-constrained vector quantization*, IEEE Trans. Acoust. Speech Signal Process. **37** (1989), no. 1, 31–42.
- [CLG89b] Philip A. Chou, Tom Lookabaugh, and Robert M. Gray, *Optimal pruning with applications to tree-structured source coding and modeling*, IEEE Trans. Inf. Theory **35** (1989), no. 2, 299–315.
- [CW96] Michael C. Chen and Alan N. Wilson Jr., *Rate-distortion optimal motion estimation algorithm for video coding*, Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, vol. 4, May 1996, pp. 2096–2099.
- [CW98] Michael C. Chen and Alan N. Wilson Jr., *Rate-distortion optimal motion estimation algorithm for motion-compensated transform video coding*, IEEE Trans. Circuits Syst. Video Technol. **8** (1998), no. 2, 147–158.
- [DFL98] Eric Dubois, Jamal Fadli, and Daniel Lauzon, *Vector quantization and coding of the chromatic information in an image*, Proc. SPIE Visual Communications and Image Process., vol. 3309, January 1998, pp. 832–841.
- [dGP95] Diego P. de Garrido and William A. Pearlman, *Conditional entropy-constrained vector quantization : High-rate theory and design algorithms*, IEEE Trans. Inf. Theory **41** (1995), no. 4, 901–916.
- [DMK95] Viet-Nam Dang, Abdul-Reza Mansouri, and Janusz Konrad, *Motion estimation for region-based video coding*, Proc. IEEE Int. Conf. Image Processing (Washington), vol. 2, October 1995, pp. 189–192.
- [Dub92] Eric Dubois, *Motion-compensated filtering of time-varying images*, Multidimensional Systems and Signal Processing **3** (1992), 211–239.

- [ECG99] Michelle Effros, Philip A. Chou, and Robert M. Gray, *Weighted universal image compression*, IEEE Trans. Image Process. **8** (1999), no. 10, 1317–1329.
- [ECH95] Touradj Ebrahimi, Homer Chen, and Barry G. Haskell, *Joint motion estimation and segmentation for very low bitrate video coding*, Proc. SPIE Visual Communications and Image Process., vol. 2501, 1995, pp. 787–797.
- [Foo95] Majid Foodeei, *Entropy-constrained recursive vector quantization and application to image sequence coding*, Ph.D. thesis, McGill University, Montreal, 1995.
- [GG84] Stuart Geman and Donald Geman, *Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images*, IEEE Trans. Pattern Anal. Machine Intell. **PAMI-6** (1984), no. 6, 721–741.
- [GG91] Allen Gersho and Robert M. Gray, *Vector quantization and signal compression*, Kluwer Academic Publishers, 1991.
- [Gir87] B. Girod, *The efficiency of motion-compensating prediction for hybrid coding of video sequences*, IEEE J. Sel. Areas Commun. **SAC-5** (1987), no. 7, 1140–1154.
- [Gir93] B. Girod, *Motion-compensating prediction with fractional-pel accuracy*, IEEE Trans. Communications **41** (1993), no. 4, 604–612.
- [Gir94] B. Girod, *Rate-constrained motion estimation*, Proc. SPIE Visual Communications and Image Process., vol. 2308, September 1994, pp. 1026–1034.
- [Gir00] B. Girod, *Efficiency analysis of multi-hypothesis motion-compensated prediction for video coding*, IEEE Trans. Image Process. **9** (2000), no. 2, 173–183.
- [GL85] Gene H. Golub and Charles F. Van Loan, *Matrix computations (iterative methods for linear systems)*, Johns Hopkins series in mathematical sciences, vol. 3, ch. 10, pp. 352–361, Johns Hopkins University Press, 1985.

- [Gu94] Chuang Gu, *3D contour image coding based on morphological filters and motion estimation*, Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, vol. 5, April 1994, pp. 277–280.
- [GWS⁺98] B. Girod, T. Wiegand, E. Steinbach, M. Flierl, and X. Zhang, *High-order motion compensation for low bit-rate video*, Proc. European Signal Processing Conference, September 1998, pp. 253–256.
- [Ham92] Horst Hampel et al, *Technical features of the JBIG standard for progressive bi-level image compression*, Signal Process., Image Commun. **4** (1992), no. 2, 103–111.
- [HB98] R. Harisdasan and J.S. Baras, *Accurate segmentation and estimation of parametric motion fields for object-based video coding using mean field theory*, Proc. SPIE Visual Communications and Image Process., January 1998, pp. 361–369.
- [Hor86] B.K.P. Horn, *Matrix computations (useful mathematical techniques)*, ch. Appendix, pp. 453–474, MIT Press, 1986.
- [HS81] B.K.P. Horn and B.G. Schunck, *Determining optical flow*, Artificial Intelligence **17** (1981), 185–203.
- [Hua94] Chung-Lin Huang, *A new motion compensation method for image sequence coding using hierarchical grid interpolation*, IEEE Trans. Circuits Syst. Video Technol. **4** (1994), no. 1, 42–51.
- [JFB95] Rajan L. Joshi, Thomas R. Fischer, and Roberto H. Bamberger, *Lossy encoding of motion vectors using entropy-constrained vector quantization*, Proc. IEEE Int. Conf. Image Processing, vol. 3, October 1995, pp. 109–112.
- [KD91] J. Konrad and E. Dubois, *Comparison of stochastic and deterministic solution methods in bayesian estimation of 2D motion*, Image Vis. Comput. **9** (1991), 215–228.

- [KD92] J. Konrad and E. Dubois, *Bayesian estimation of motion vector fields*, IEEE Trans. Pattern Anal. Machine Intell. **14** (1992), 910–927.
- [Key81] Robert G. Keys, *Cubic convolution interpolation for digital image processing*, IEEE Trans. Acoust. Speech Signal Process. **ASSP-29** (1981), no. 6, 1153–1160.
- [KMD⁺95] Janusz Konrad, Abdul-Reza Mansouri, Eric Dubois, Viet-Nam Dang, and Jean-Bernard Chartier, *On motion modelling and estimation for very low bit rate video coding*, Proc. SPIE Visual Communications and Image Process., vol. 2501, May 1995, pp. 262–273.
- [KMW97] R. Krishnamurthy, P. Moulin, and J.W. Woods, *Compactly-encoded optical flow fields for bi-directional prediction and video coding*, Proc. SPIE Visual Communications and Image Process., February 1997, pp. 1164–1173.
- [LD97] Jungwoo Lee and Bradley W. Dickinson, *Rate-distortion optimized frame type selection for MPEG encoding*, IEEE Trans. Circuits Syst. Video Technol. **7** (1997), no. 3, 501–510.
- [LD98] D. Lauzon and E. Dubois, *Representation and estimation of motion using a dictionary of models*, Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, May 1998, pp. 2585–2588.
- [LD99] Daniel Lauzon and Eric Dubois, *Rate-constrained video coding using a flexible representation of motion*, Proc. SPIE Visual Communications and Image Process., vol. 3653, January 1999, pp. 518–527.
- [Lee95] Jungwoo Lee, *Optimal quadtree for variable block size motion estimation*, Proc. IEEE Int. Conf. Image Processing, vol. 3, October 1995, pp. 480–483.
- [Lee98] Jungwoo Lee, *Joint optimization of block size and quantization for quadtree based motion estimation*, IEEE Trans. Image Process. **7** (1998), no. 6, 909–912.

- [LS95] W. Li and E. Salari, *Successive elimination algorithm for motion estimation*, IEEE Trans. Image Process. **4** (1995), no. 1, 105–107.
- [LZ86] Abraham Lempel and Jacob Ziv, *Compression of two-dimensional data*, IEEE Trans. Inf. Theory **32** (1986), no. 1, 1–8.
- [MBW97] Alistair Moffat, T. C. Bell, and Ian H. Witten, *Lossless compression for text and images*, Int. Journal of High Speed Electronics and Systems **8** (1997), no. 1, 179–231.
- [MF97] Stéphane Mallat and Frédéric Falzon, *Understanding wavelet image compression*, Proc. SPIE, vol. 3078, April 1997, pp. 74–93.
- [MK00] Abdul-Reza Mansouri and Janusz Konrad, *Multiple motion segmentation with level sets*, IEEE Trans. Image Process. (2000), 1–21.
- [NH94] Yuchiro Nakaya and Hiroshi Harashima, *Motion compensation based on spatial transformations*, IEEE Trans. Circuits Syst. Video Technol. **4** (1994), no. 3, 339–356.
- [OR98] Antonio Ortega and Kannan Ramchandran, *Rate-distortion methods for image and video compression*, IEEE Signal Process. Mag. **15** (1998), no. 6, 23–50.
- [PC95] C.A. Papadopoulos and Trevor G. Clarkson, *Motion compensation using second-order geometric transformations*, IEEE Trans. Circuits Syst. Video Technol. **5** (1995), no. 4, 319–331.
- [RCN95] Jordi Ribas-Corbera and David L. Neuhoff, *Optimal bit allocations for lossless video coders : Motion vectors vs. difference frames*, Proc. IEEE Int. Conf. Image Processing, 1995, pp. 180–183.
- [RCN96a] Jordi Ribas-Corbera and David L. Neuhoff, *On the optimal motion vector accuracy for block-based motion-compensated video coders*, Proc. SPIE Digital Video Compression : Algorithms and Technology, vol. 2668, March 1996, pp. 302–314.

- [RCN96b] Jordi Ribas-Corbera and David L. Neuhoff, *Reducing rate/complexity in video coding by motion estimation with block adaptive accuracy*, Proc. SPIE Visual Communications and Image Process., vol. 2727, February 1996, pp. 615–624.
- [ROV94] K. Ramchandran, A. Ortega, and M. Vetterli, *Bit allocation for dependant quantization with applications to multiresolution and MPEG video coders*, IEEE Trans. Image Process. **3** (1994), no. 5, 533–545.
- [SG88] Yair Shoham and Allen Gersho, *Efficient bit allocation for an arbitrary set of quantizers*, IEEE Trans. Acoust. Speech Signal Process. **36** (1988), no. 9, 1445–1453.
- [Sha48] Claude E. Shannon, *A mathematical theory of communication*, Bell System Technical Journal **27** (1948), 379–423, 623–656.
- [Sha93] Jerome M. Shapiro, *Embedded image coding using zerotress of wavelet coefficients*, IEEE Trans. Signal Process. **41** (1993), no. 12, 3445–3462.
- [SK95a] Christoph Stiller and Janusz Konrad, *Eigentransforms for region-based image processing*, Proc. Int. Conf. on Consumer Electronics (Chicago), June 1995, pp. 286–287.
- [SK95b] Christoph Stiller and Janusz Konrad, *Region-adaptive transform based on a stochastic model*, Proc. IEEE Int. Conf. Image Processing (Washington), vol. 2, October 1995, pp. 264–267.
- [SK97] Guido M. Schuster and Aggelos K. Katsaggelos, *Optimal decomposition for quad-trees with leaf dependencies*, Proc. SPIE Visual Communications and Image Process., vol. 3024, February 1997, pp. 59–70.
- [SK98] Guido M. Schuster and Aggelos K. Katsaggelos, *An optimal quadtree-based motion estimation and motion-compensation interpolation scheme for video compression*, IEEE Trans. Image Process. **7** (1998), no. 11, 1505–1523.
- [SK99] Christoph Stiller and Janusz Konrad, *Estimating motion in image sequences*, IEEE Signal Process. Mag. **16** (1999), 70–91.

- [SKJ97] Huifang Sun, Wilson Kwok, and C. H. John Ju, *MPEG coding performance improvement by jointly optimizing code mode decisions and rate control*, IEEE Trans. Circuits Syst. Video Technol. **7** (1997), no. 3, 449–458.
- [SLZ90] Dafna Sheinwald, Abraham Lempel, and Jacob Ziv, *Two dimensional encoding by finite-state encoders*, IEEE Trans. Communications **38** (1990), no. 3, 341–347.
- [SMK98] Guido M. Schuster, Gerry Melnikov, and Aggelos K. Katsaggelos, *Operationally optimal vertex-based shape coding*, IEEE Signal Process. Mag. **15** (1998), no. 6, 91–108.
- [SP96a] Amir Said and William A. Pearlman, *An image multiresolution representation for lossless and lossy image compression*, IEEE Trans. Image Process. **5** (1996), 1303–1310.
- [SP96b] Amir Said and William A. Pearlman, *A new fast and efficient image codec based on set partitioning in hierarchical trees*, IEEE Trans. Circuits Syst. Video Technol. **6** (1996), no. 3, 1303–1310.
- [SSG96] K.W. Stuhlmüller, A. Salai, and B. Girod, *Rate-constrained contour representation for region based motion compensation*, Proc. SPIE Visual Communications and Image Process., vol. 2727, March 1996, pp. 344–345.
- [Sti94] Christoph Stiller, *Object-oriented video coding employing dense motion fields*, Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, vol. 5, 1994, pp. 273–276.
- [SW98] Gary J. Sullivan and Thomas Wiegand, *Rate-distortion optimization for video compression*, IEEE Signal Process. Mag. **15** (1998), no. 6, 74–90.
- [TK95] Damon L. Tull and Aggelos K. Katsaggelos, *Regularized motion estimation using robust entropy functionals*, Proc. IEEE Int. Conf. Image Processing, vol. 3, October 1995, pp. 212–215.

- [Wan98] Demin Wang, *Unsupervised video segmentation based on watersheds and temporal tracking*, IEEE Trans. Circuits Syst. Video Technol. **8** (1998), no. 5, 539–546.
- [WL00a] Demin Wang and Daniel Lauzon, *Block/object-based algorithm for estimating true motion fields*, Proc. SPIE Visual Communications and Image Process., vol. 4067, June 2000, pp. 1374–1383.
- [WL00b] Demin Wang and Daniel Lauzon, *A hybrid algorithm for estimating true motions fields*, Optical Engineering **39** (2000), no. 11, 2876–2881.
- [WLG98] T.W. Wiegand, Bo Lincoln, and B. Girod, *Fast search for long-term memory motion-compensated prediction*, Proc. IEEE Int. Conf. Image Processing, vol. 3, October 1998, pp. 619–622.
- [WNC87] Ian H. Witten, Radford M. Neal, and John G. Cleary, *Arithmetic coding for data compression*, Communications of the ACM **30** (1987), no. 6, 520–540.
- [WSSG98] T.W. Wiegand, E. Steinbach, A. Stensrud, and B. Girod, *Multiple reference picture video coding using polynomial motion models*, Proc. SPIE Visual Communications and Image Process., January 1998, pp. 134–135.
- [Wu96] Xiaolin Wu, *An algorithmic study on lossless image compression*, Data Compression Conference, 1996, pp. 150–159.
- [Wu97] Xiaolin Wu, *Lossless compression of continuous-tone images via context selection, quantization and modeling*, IEEE Trans. Image Process. **6** (1997), no. 5, 656–664.
- [WWW99] Xiaolin Wu, Jiang Wen, and Wing Hung Wong, *Conditional entropy coding of VQ indexes for image compression*, IEEE Trans. Image Process. **8** (1999), no. 8, 1005–1013.
- [YC95] Guo Yao Yu and Cheng-Tie Chen, *Two-dimensional motion vector coding for low bitrate videophone applications*, Proc. IEEE Int. Conf. Image Processing, 1995, pp. 414–417.

- [YVK95] Joseph Yeh, Martin Vetterli, and Masoud Khansari, *Motion compensation of motion vectors*, Proc. IEEE Int. Conf. Image Processing, October 1995, pp. 574–569.
- [ZB95] Heyun Zheng and Steven D. Blostein, *Motion-based object segmentation and estimation using the mdl principle*, IEEE Trans. Image Process. **4** (1995), no. 9, 1223–1234.