École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

présentée pour obtenir le grade de docteur

de TÉLÉCOM ParisTech

Spécialité : Signal et Images

Thomas MAUGEY

Codage vidéo distribué de séquences multi-vues.

–

Distributed video coding of multiview sequences.

Soutenue le 18 novembre 2010 devant le jury composé de

Marc Antonini	Président
Christine Guillemot	Rapporteurs
Pascal Frossard	
Michel Kieffer	Examinateur
Béatrice Pesquet-Popescu	Directrice de thèse
Marco Cagnazzo	Co-encadrant

"Juste voguait par là, le bateau des copains,
je me suis accroché bien fort à ce grappin.
Et par enchantement tout fut régénéré,
l'espérance cessa d'être désespérée."

Georges Brassens

Je pense en premier lieu à tous ceux qui m'ont accompagné scientifiquement durant ces trois ans et demi à TELECOM *ParisTech. Je remercie les membres de mon jury pour la qualité de leur évaluation, de leurs retours et de leurs conseils: Marc Antonini, président, Christine Guillemot et Pascal Frossard, rapporteurs, Michel Kieffer, examinateur. Je tiens, bien sûr, à témoigner de toute ma reconnaissance à mes deux directeurs de thèses: Marco Cagnazzo et Béatrice Pesquet-Popescu. J'ai particulièrement apprécié la confiance qu'ils m'ont donnée, les discussions nombreuses et constructives que nous avons eues et la patience qu'ils ont montrée lors des différentes relectures et corrections. Je pense aussi à Christophe Tillier qui m'a fait rentrer à* TELECOM *ParisTech et qui a su m'enseigner les ficelles du codage vidéo distribué durant mon stage de Master. Je remercie également les membres du projet* ESSOR *et plus particulièrement Michel, Christine, Marc, Olivier, Marie-Andrée et Cagatay. Je tiens aussi à remercier Joumana et Charles pour la collaboration fort enrichissante que nous avons menée. Je n'oublierai pas non plus le formidable accueil qu'ils m'ont réservé lors de ma venue au Liban (je pense aussi à la famille de Charles). Je remercie aussi mes collègues de l'*EPFL *pour leur accueil parmi eux.*

Il n'aurait été possible d'effectuer cette thèse dans d'aussi bonnes conditions sans l'aide précieuse de Laurence au secrétariat, sans la gentillesse et la douceur de Clara et Maryse à l'accueil, sans les discussions enflammées avec Auguste à la sécurité. Un remerciement très spécial à Fabrice pour son aide technique mais surtout psychologique. Je n'oublierai pas ces discussions et fous rires avec lui. Qu'auraient été ces trois années sans l'aide et le soutien (scientifique et moral) des collègues. Un grand merci à ceux qui m'ont si bien accueillis et appris la vie de jeune chercheur lors de mon arrivée : Maria, Ismaël, Aurélia, Téodora, Tyze, Lionel, Jean, Valentin. Une pensée aussi à deux de mes co-auteurs qui, entre autres, m'ont accompagné durant les longues soirées de soumission d'articles : Thomas et Jérôme (que je remercie aussi pour toute son aide et soutien). Je pense aussi aux autres collègues et leur souhaite beaucoup de réussite: Mounir, Brahim, Claudio, Manel, Giovanni, Rafael, Irina, Eli, Abdel Bassir, Erica et enfin Valentina. Je retiendrai enfin les liens d'amitiés développés avec certains, qui ont su m'apporter réconfort et évasion.

Je tiens à remercier du fond du coeur tous ceux qui sont venus assister à la soutenance de ma thèse.

J'ai eu la chance durant ces trois ans de pouvoir compter sur une famille présente, réconfortante, attentive, aimante et motivante. Alors, merci à mes parents, à mon frère Mathieu et ma soeur Anaïs pour leur amour et la douceur des moments partagés. Merci à la famille plus élargie, merci à tous mes cousins cousines et plus particulièrement à Julia, Carol, Florie et Guillaume pour leur présence, et leur soutien. Je pense aussi très fort à mes grand-parents et à ma grande tante Germaine. Merci à la famille encore plus élargie : à Didier, à Alain.

Une famille ne me suffisant apparemment pas, j'ai eu la chance d'en posséder d'autres. Je ne remercierai jamais assez Sit, Astrid, Louve, ainsi que Lem, Zazard, Briard, Yann, Adrien, Marc et Audrey. Je leur dois une fidèle amitié, belle, sincère et réconfortante sur laquelle j'ai pu douillettement me reposer lors de ces trois années parfois difficiles.

J'ai également eu la chance de pouvoir profiter de mes soirées et week-end au sein des EEUDF *pour m'évader. Merci à toute l'ER et principalement à Pierre, Laurent, Hervé, Elodie, Laure, Florence et Alexis. Un merci très spécial à Fanny dont l'oreille m'a été fort utile et reposante durant ces trois ans. Merci également à tous ceux que j'ai eu la chance de rencontrer au détour de stages BAFA ou BAFD, qui ont su m'aider à leur manière et envers qui je suis très reconnaissant : Amélie, Thylacine, David, Sarah, Céline et enfin Magali pour ces discussions enflammées et son écoute.*

Puis il y a les vieux amis, Guillaume et Laura qui me suivent depuis tout petit et dont je considère le jugement comme primordial voire fondateur. Merci à Martin et Alexis, merci à Fred et Marie, à la clique du petit balcon (Cédric, Flo, Hashour, Malikette et Mumu), et à Pierre, Fabien, Rémi et Camille, enfin à Roman re-rencontré par hasard et qui a formidablement bien accompagné les dimanches de ma fin de thèse.

Un profond merci à Yalii, pour avoir été là, pour m'avoir soutenu, remis dans la réalité par moment et pour m'avoir regardé de cette admirable façon ce jour-là. Je n'oublierai pas.

La fin de ces longs remerciements est destinée à un ancien étudiant de ma promo, chanteur de mon groupe, puis collègue et désormais ami. Débutées ensembles, nos deux thèses nous ont permis de découvrir à quel point nous étions faits pour nous rencontrer. Plus qu'un compagnon de chemin, Laurent a été pour moi un élément indispensable de mon quotidien durant ces trois années. Je dirais même qu'il fut à certains moments le seul à réellement me comprendre, à réellement mesurer mes doutes, et savoir quoi me dire pour me faire repartir de l'avant. Merci, donc pour tous ces débriefings du lundi, merci pour ces fous rires. En espérant pouvoir un jour réaliser notre rêve, celui d'enseigner ensemble, je tiens à témoigner en ces lignes de ma profonde reconnaissance pour le rôle primordial qu'il a joué dans la longue élaboration de cette thèse.

Résumé

Depuis 2002, le codage vidéo distribué a connu un véritable essor de par ses résultats théoriques séduisants, et ses applications potentielles attractives. En effet, avec ce mode de compression, toute comparaison inter-image est transférée au décodeur, ce qui implique une baisse considérable de la complexité à l'encodeur, et de plus, un encodage indépendant des caméras dans le cas de compression multi-vues. Cette thèse a pour but de proposer de nouvelles solutions dans le domaine du codage vidéo distribué, et particulièrement dans son application aux systèmes multi-caméra. Ces contributions se présentent sous plusieurs aspects : un nouveau modèle débit-distorsion et sa mise en pratique sur trois problématiques, de nouvelles méthodes de construction de l'information adjacente et enfin une étude approfondie du décodeur des trames Wyner-Ziv. Ces nouvelles approches ont toutes pour but d'améliorer les performances débit-distorsion ou de permettre une compréhension plus précise du comportement du codeur. Celles-ci sont exposées en détail dans ce manuscrit avec au préalable une explication complète du contexte dans lequel elles s'inscrivent.

Abstract

Since 2002, distributed video coding has become a major paradigm, because of its attractive theoretical results, and its promising target applications. Indeed, in such a compression system, all inter frame comparison is shifted from the encoder to the decoder, which implies an important complexity reduction at the encoder, and moreover, an independent encoding of the camera in case of multiview compression. This thesis aims at proposing new solutions for distributed video coding, and especially within multi-camera setting. These contributions handle several aspects of distributed video coding paradigm as: a new rate-distortion model and its applications, novel side information generation techniques and finally a detailed study of the Wyner-Ziv decoder. All these new approaches aim at enhancing the rate-distortion performance or at leading to a better comprehension of the coder behavior. These ones are explained in detail in this manuscript preceded by a complete overview of their context.

Résumé en français

Introduction

La compression vidéo est un enjeu de recherche, qui depuis des décennies, mobilise de nombreuses équipes et de nombreux industriels. Depuis son objectif initial qui consistait à simplement diminuer de plus en plus le débit nécessaire à la description d'un flux vidéo, de nombreuses problématiques ont émergé, avec pour seules différences des conditions de transmission, de matériel, ainsi que de puissance des codeurs/décodeurs. En effet, si pour chacun des paradigmes s'inscrivant dans le domaine général de la compression vidéo, le but reste d'améliorer les performances du compromis haute qualité de décodage et faible débit, il n'en reste pas moins que les conditions dans lesquelles s'opère cette compression influent nettement sur les objectifs plus précis, et les techniques employées. Par exemple, le schéma de codage ne sera pas le même s'il s'inscrit dans une transmission sur un réseau, sur un canal bruité ou non bruité. De manière similaire, les techniques de compression employées différeront selon la puissance des encodeurs et des décodeurs, ou selon s'il y a une ou plusieurs caméras.

La compression vidéo dite *classique* (car plus courante) s'emploie à extraire la corrélation entre les images à l'encodeur. C'est ainsi qu'elle fait appel à des techniques complexes (en terme de puissance de calcul) telles que l'estimation de mouvement (ou de disparité dans le cas de séquences multivues) pour diminuer la quantité d'information à transmettre au décodage. Ce schéma de compression est parfaitement adapté aux conditions matérielles suivantes: une compression sur une station à forte capacité de calcul, et un décodage léger sur des systèmes à faible puissance (platine DVD, diffusion de la télévision, etc.). Or, de nos jours, bien que ce type de configuration reste très utilisé, de nouveaux besoin ont émergé ces dernières années. En effet, de plus en plus de systèmes légers se sont doté de matériel de capture, et ont ainsi eu le besoin de compresser des séquences vidéos (par exemple des téléphones portables). En outre, de plus en plus de systèmes employant des réseaux de caméras (comme la vidéo surveillance) nécessitent une compression légère et surtout sans communication entre les caméras (obligatoire avec le codage classique si l'on veut exploiter la corrélation entre les caméras).

C'est à partir de ces types de besoins qu'est né, en 2002, le codage vidéo dit *distribué*, dont le principe est de transférer au décodeur tout type de calcul visant à une quelconque comparaison inter-image. Cette idée provient de résultats théoriques publiés 30 ans plus tôt par Slepian et Wolf d'une part, et Wyner et Ziv d'autre part, qui prouvent que sous certaines conditions, l'encodage de deux sources corrélées peut se faire conjointement ou indépendamment sans qu'il n'y ait de perte d'efficacité de transmission à partir du moment ou le décodage est, lui, effectué conjointement.

Ces séduisants résultats théoriques encouragèrent de nombreuses équipes de chercheurs à se lancer dans le développement de schémas de codage vidéo distribué avec comme but (théoriquement possible) d'égaler les performances des schémas classiques tels que MPEG-x, H.263 puis H.264, etc. Seulement, même si le codage vidéo distribué a connu des débuts prometteurs, les performances débit-distorsion des codeurs actuels sont encore loin du but. En effet, un certain nombre d'hypothèses des théorèmes des années 1970 ne sont pas forcément respectées et limitent un peu la progression des performances de codage. Il n'en reste pas moins que la marge de progression des codeurs vidéo distribués est encore grande et nombreux de leurs modules peuvent encore être améliorés.

Dans le cadre du projet européen DISCOVER, un certain nombre de laboratoires ont développé un schéma complet de codage vidéo distribué qui est actuellement l'un des plus efficaces et l'un des plus populaires. Ce schéma constituera le point de départ de la plupart des travaux présentés dans cette thèse, et c'est pourquoi nous en dégageons ici les principales problématiques. Les images de la séquence sont réparties en deux types, les trames clefs et les trames Wyner-Ziv (WZ), réparties selon la structure suivante (répétée tout au long de la séquence) : une trame clef suivie de n trames WZ. Les images clefs sont encodées et décodées de manière indépendante grâce à des codecs de type Intra, tels que H.264 Intra ou JPEG2000. Celles-ci sont utilisées au décodeur pour générer une estimation des trames WZ appelée information adjacente. De leur côté, les trames WZ sont également encodées indépendamment, et subissent le traitement classique de compression de données, à savoir une transformation suivie d'une quantification. Ensuite, à la place du codeur entropique (usuellement utilisé pour les schémas de compression classiques) le flux résultant de la quantification est traité par un codeur canal (LDPC ou turbodécodeur), celui-ci produisant, par nature, un flux systématique (une version de l'information en entrée) et un flux de parité (une redondance utilisée pour corriger les erreurs de transmission). L'astuce de ce type de schéma est de ne pas transmettre le flux systématique et de le substituer au décodeur par l'information adjacente générée grâce aux trames clefs. Ainsi, l'information de parité, initialement destinée à corriger les erreurs de canal, est transmise ici dans le but d'annuler les erreurs d'estimation. Le flux WZ alors reconstruit est finalement projeté dans le domaine pixel.

L'astuce de la compression utilisant des codeurs canal est celle qui fait l'originalité et l'attractivité du codage vidéo distribué, mais c'est aussi celle qui implique le plus d'éléments limitant et le plus de travaux de recherche. Premièrement, elle implique de connaître la corrélation de l'information adjacente et de la trame WZ originale, or ni à l'encodeur, ni au décodeur ces deux informations sont disponibles en même temps. De plus, l'encodeur doit savoir la quantité exacte d'information de parité à envoyer. C'est pourquoi, le schéma de codage DISCOVER (et quasiment toutes ses variantes) effectue un décodage progressif avec un canal de retour pour demander au fur et à mesure à l'encodeur d'envoyer plus d'information parité. C'est l'une des plus grosses limitations de ces schéma, car elle implique un décodage en temps réel, difficilement réalisable.

Le second élément déterminant de ce type de schéma est la génération d'information adjacente fondée sur les trames clefs. Les performances de codage dépendent fortement de la qualité de l'estimation de la trame WZ. C'est pourquoi de nombreuses recherches s'attellent à améliorer la précision de l'information adjacente en proposant des méthodes efficaces d'estimation de mouvement ou de disparité notamment.

Les travaux menés durant cette thèse nous ont conduit à nous intéresser à plusieurs des problématiques du codage vidéo distribué. Tout d'abord, nous avions pour objectif d'étudier précisément les conditions d'extension du codage vidéo distribué au cas multivue pour lequel de nouvelles questions se posent, comme la disposition stratégique des trames clefs et des trames WZ dans le plan temps-vues, ou bien la manière de générer des estimations intervue, et de les fusionner avec l'estimation temporelle afin d'obtenir une unique information adjacente. Tout en proposant des solutions à ces différents enjeux, nous avons été amenés à nous pencher sur des problématiques du codage vidéo distribué en général (non spécifiques au multivue) comme une amélioration de l'interpolation temporelle, le raffinement du modèle de bruit de corrélation au turbodécodeur, la suppression du canal de retour, ou encore l'étude de métriques servant à estimer la qualité de l'information adjacente. De plus, nous nous sommes penchés sur des schémas de codage vidéo distribué différents de Discover. Ainsi, nous avons proposé une nouvelle approche pour les schémas utilisant de l'information de hachage. En outre, dans le cadre du projet ANR Essor nous avons développé en collaboration avec le LSS, l'IRISA et I3S un codeur s'inspirant de la structure de Discover mais adoptant une approche de codage en ondelettes pour les trames clefs et WZ.

Ainsi, dans le manuscrit qui suit, nous présentons nos contributions, après avoir détaillé leur contexte et objectif. Celles-ci sont organisées en trois parties correspondant chacune à une thématique générale dans laquelle s'inscrivent les solutions proposées. Une première partie traite de tout ce qui vise à améliorer la compréhension du codeur en général, et des performances débit-distorsion en particulier. Dans une seconde partie, nous nous penchons sur tout ce qui a trait avec l'information adjacente, et enfin dans une dernière partie nous effectuons un zoom sur le turbodécodeur et ses problématiques. Voici le détail des différents chapitres composants ce manuscrit.

Chapitre 1 - l'état de l'art du codage distribué : nous présentons les origines du codage vidéo distribué à travers l'étude rapide des méthodes existantes de codage source distribué, et de leur deux principales extensions à la vidéo. De plus, nous entrons en détail dans le fonctionnement du codeur Discover et présenterons les différentes problématiques qui en découlent. Ce chapitre ne présente pas un état de l'art détaillé de chacun des modules car ceux-ci sont proposés plus tard dans les chapitres appropriés.

Partie 1 - Proposition et application d'un modèle débit-distorsion : Dans cette partie, nous nous intéressons au comportement général des performances débit-distorsion du schéma de codage distribué. En se fondant sur un modèle débit distorsion original, nous étudions plus précisément l'entrée du codeur (et la classification des types d'images), puis la sortie avec le phénomène de propagation d'erreurs en cas de perte d'image. Enfin, nous nous penchons sur l'étude de la suppression du canal de retour.

Chapitre 2 - un nouveau modèle débit-distorsion : nous présentons ici une étude originale visant à modéliser l'erreur d'estimation de la trame WZ au décodeur. L'expression obtenue comporte une structure très simple qui sépare l'erreur provenant de la quantification des trames de référence, et l'erreur provenant de l'estimation de mouvement. Ce modèle suppose un certains nombre d'hypothèses, qui seront testées dans ce chapitre.

Chapitre 3 - Application : dans ce chapitre nous décrivons trois problématiques pour lesquelles nous avons eu recours au modèle proposé. La première concerne la classification des images à l'entrée du schéma de codage. Ainsi, nous détaillons les classifications

existantes et en proposons une comportant une nombre plus réduit de trame clefs, réduisant ainsi la complexité de l'encodeur. Grâce au modèle débit-distorsion proposé, nous établirons une stratégie de décodage optimale (ordre de traitement des trames au décodeur). Ensuite, nous nous pencherons sur le phénomène de propagation d'erreurs dans le cas d'une perte d'image au moment de la transmission d'une vidéo monovue. Nous étudions l'importance des images en fonction de leur position dans l'ordre de décodage, et nous apercevrons d'un certains nombre de problématiques liées au contrôle du débit à l'encodeur comme le fait de ne pas allouer le même débit aux trames WZ selon la position qu'elles occupent dans la séquence. Enfin, nous proposons un schéma original de suppression du canal de retour se fondant sur le modèle de distorsion proposé afin d'allouer le débit par trame, et en le répartissant entre les plans de bits des différentes bandes en fonction d'un calcul utilisant la distance de Hamming.

Partie 2 - génération de l'information adjacente : dans cette partie nous nous intéressons exclusivement à l'estimation de la trame WZ au décodeur. Après avoir effectué une revue de littérature précise des méthodes existantes, nous nous présentons premièrement l'algorithme d'interpolation développé au sein du projet ESSOR. Puis nous détaillons les méthodes d'interpolation denses (un vecteur par pixel) proposées ainsi que nos méthodes de fusion de l'estimation temporelle et intervue. Enfin, nous présentons notre approche originale de schéma à base d'information de hachage.

Chapitre 4 - état de l'art : nous présentons ici en détail les différentes problématiques liées à l'information adjacente qui sont les méthodes d'estimations (interpolation, extrapolation, etc.), puis leur fusion dans le cas de multiples estimations, et enfin les schéma à base d'information de hachage existants.

Chapitre 5 - interpolation ESSOR : ce chapitre a pour but de présenter la méthode d'interpolation proposée dans le cadre du projet ESSOR. Nous détaillons également le codeur dans lequel cet algorithme s'inscrit, et montrerons certains résultats débit-distorsion.

Chapitre 6 - méthodes denses : fondé sur l'idée que l'économie du nombre de vecteurs servant à effectuer les interpolations au décodeur n'était pas justifiée (car ces vecteurs ne sont en fait pas transmis comme ce serait le cas dans un schéma classique), et qu'il était donc possible de décrire le mouvement grâce à des champs denses (un vecteur par pixel) nous avons proposé une famille de méthodes de raffinement du champ, en se fondant sur la structure de la méthode d'interpolation de DISCOVER, et en adaptant deux techniques de raffinement existantes : l'algorithme de Cafforio-Rocca [Cafforio, Rocca, 1983] et de Miled [Miled *et al.*, 2009] (fondé sur l'étude des variations totales). Enfin, dans ce chapitre nous proposons trois méthodes de fusion originales, dans le sens où elles adoptent une approche linéaire (combinaison linéaire des candidats) alors que la litérature n'effectue que des fusions binaires (l'un ou l'autre des candidats).

Chapitre 7 - schéma à base d'information de hachage : en partant du principe que le décodeur n'a pas toutes les informations nécessaires à l'estimation parfaite de la trame WZ, certaines solutions proposent de transmettre ce qu'on appelle de l'information de hachage, et qui correspond à une description localisée et bien choisie de l'image WZ, de manière à améliorer son estimation au décodage. Dans ce chapitre nous proposons une nouvelle approche pour générer et sélectionner l'information de hachage, et en outre nous proposons détendre l'algorithme proposé par Yaacoub et al [Yaacoub *et al.*, 2009a] pour la génération d'information adjacente dans le cas multivue.

Partie 3 - zoom au niveau du turbodécodeur : Dans cette partie, nous nous penchons sur deux problématiques liées au tubodécodeur. Premièrement, nous proposons un raffinement de la modélisation du bruit de corrélation, et enfin, nous nous intéressons aux métriques servant à estimer la qualité de l'information adjacente.
Chapitre 8 - modélisation du bruit de corrélation : dans ce chapitre nous présentons une revue détaillée des méthodes existantes visant à modéliser le bruit de corrélation. D'après cette revue de littérature, nous pouvons constater que plus le modèle est fin (et proche de la vraie distribution de l'erreur), plus les performances sont bonnes. Ainsi, nous avons proposé d'utiliser le modèle gaussien généralisé plutôt que la trop peu générale laplacienne unanimement adoptée. Les résultats obtenus sont mitigés. Bien que dans de nombreux cas, le raffinement proposé par la gaussienne généralisée présente des résultats très acceptables, il existe certains cas pour lesquels les performances restent inchangés. Nous effectuons donc dans ce chapitre une étude un peu plus poussée de la modélisation du bruit de corrélation afin de mieux comprendre et analyser les résultats obtenus.
Chapitre 9 - étude de la qualité de l'information adjacente : lorsqu'une méthode de génération d'information adjacente est testée, elle est souvent évaluée grâce au PSNR. Or Kubasov [Kubasov, 2008] a montré que cette métrique pouvait par moment donner une idée erronée de cette qualité. Dans ce chapitre, nous proposons d'étendre l'étude initiée par celui-ci. Ainsi, nous tentons de comprendre dans quelles situations le PSNR semble adéquat, et dans quels cas cette mesure peut présenter des limites de fiabilité. De plus, nous testons pour chacun de ces cas de figure la fiabilité d'autres mesures, plus proche du comportement du turbodécodeur.

Annexe - Utilisation des méthodes d'estimation de disparité pour le compressed sensing appliqué aux images multivues : j'ai également été amené durant mon doctorat, à travailler sur d'autres sujets annexes que je n'intègre pas dans ma thèse car trop éloignés du domaine du codage vidéo distribué. Cependant, ces travaux en parallèles sont liés avec l'approche distribuée par le fait qu'ils traitent d'un autre sujet très en vogue de nos jours: le compressed sensing, que nous avons proposé d'étendre à des images et vidéos multivues en appliquant certaines méthodes d'estimation de disparité traitées dans ce manuscrit. Vous trouvez l'ensemble des articles publiés dans cette annexe.

Afin d'implanter et évaluer les contributions ci-dessus, nous avons été amené à développer d'une part une extension au multivue du codeur Discover, et d'autre part un codeur complet basé ondelettes dans le cadre du projet Essor.
De plus, nous précisons que cette thèse s'est inscrite dans le cadre de deux projets : Essor, projet ANR constitué du LSS, de l'IRISA, de I3S et de Telecom ParisTech ainsi que Cedre, projet franco libanais en collaboration avec l'université Saint-Esprit de Kaslik.

Dans ce résumé, nous présenterons de manière synthétique l'ensemble des contributions développées durant la thèse.

Etat de l'art du codage vidéo distribué

Résumé du chapitre 1 du manuscrit de thèse.

Il est avant tout nécessaire de faire un bref historique rappelant les origines et fonde-

ments d'une telle approche dans le codage vidéo. Le problème à résoudre est de transmettre l'information générée par deux sources corrélées, X et Y, sur un canal avec les débits, R_X et R_Y, les plus faibles possibles. Au décodeur, les informations reçues, $\hat{X}$ et $\hat{Y}$ doivent également présenter la plus grande ressemblance avec l'information envoyée, et ainsi minimiser les distorsions $d(X, \hat{X})$ et $d(Y, \hat{Y})$. En 1973, Slepian et Wolf [Slepian, Wolf, 1973] étudièrent les débits minimum nécessaires à la transmission, dans le cas d'une distorsion nulle, et pour plusieurs cas de figure. Deux d'entre eux s'avèrent être les points de départs de ce qu'on appellera plus tard le codage distribué. La première configuration est celle dans laquelle, à l'encodeur comme au décodeur, le codage se fait avec une pleine connaissance de l'autre source. Autrement dit, on encode et on décode X et Y conjointement. Sous ces conditions, le débit minimum requis est $R_X + R_Y = H(X, Y)$ où $H(X, Y)$ est l'entropie conjointe des deux sources. Slepian et Wolf prouvent que ce résultat, bien connu dans cette configuration, est le même que dans le deuxième cas de figure nous intéressant ici, correspondant à la situation dans laquelle l'encodage se fait *indépendamment* (le décodage étant encore conjoint). Autrement dit, encoder des sources indépendamment plutôt que conjointement ne dégrade pas les performances tant que le décodage se fait conjointement. En 1976, Wyner et Ziv [Wyner, Ziv, 1976] étendirent ce résultat au cas d'une transmission avec perte (où $d(Y, \hat{Y}) \neq 0$).

Il fallut attendre presque trente ans avant que ces résultats théoriques prometteurs soient mis en pratique en codage vidéo. Pourtant, ils apportent une approche nouvelle adaptée à des problématiques réelles et évidentes. Depuis quelques années, la compression vidéo doit de plus en plus s'adapter à son support. Plus précisément, les mobiles ou tout autres caméras légères ne supportent pas tous les calculs que les codages usuels requièrent pour obtenir de bonnes performances. En effet, l'extraction de la corrélation entre trames se fait principalement par de l'estimation de mouvement entre images, et c'est celle-ci qui est à l'origine de la plus grande partie de la complexité des codeurs comme H.26x. En supprimant cette extraction de mouvement réalisée à l'encodeur, on peut considérablement réduire la puissance de calcul requise tout en ne dégradant théoriquement pas les performances. Les premières solutions de ce qu'on appellera le *codage vidéo distribué*, arrivèrent au début des années 2000. Deux solutions furent à l'époque proposées: PRISM [Puri, Ramchandran, 2003] et le codeur de Standford [Aaron *et al.*, 2002; Girod *et al.*, 2005]. Dans la thèse et donc dans ce résumé nous nous pencherons principalement sur la deuxième des solutions dont le schéma, représenté dans figure 1 est le suivant : la séquence vidéo est divisée en deux ensembles dont les éléments sont extraits alternativement de la vidéo afin d'augmenter la corrélation entre ceux-ci. Les *trames clefs* (TC) constituent le premier ensemble dont les images sont codées indépendamment entre elles avec un codeur intra classique (JPEG, H.26x Intra, JPEG 2000,...). Le second ensemble est composé des *trames Wyner Ziv* (TWZ). Celles-ci sont d'abord projetées dans le domaine transformé (ondelettes ou cosinus discret principalement), puis quantifiées et enfin encodées grâce à un codeur canal (turbocode ou LDCP). Ces types de codes produisent ce qu'on appelle l'information systématique (qui est la copie de l'entrée) et une information de parité qui est l'information redondante capable au décodeur de corriger les erreurs intervenues sur l'information systématique. Dans le schéma de codage vidéo distribué de Stanford seule l'information de parité est transmise (partiellement) au décodeur. L'information systématique est remplacée par une estimation de la TWZ correspondante. Cette estimation est appelée l'*information adjacente* (IA) et est générée grâce aux TC déjà décodées. Ainsi,

l'hypothèse sous-jacente est que l'erreur d'estimation est assimilable à une erreur de transmission canal. Une fois l'IA corrigée par l'information de parité, la trame est projetée dans le domaine spatial. Ce schéma de codage fut également la base du projet européen DISCOVER [Guillemot *et al.*, 2007].

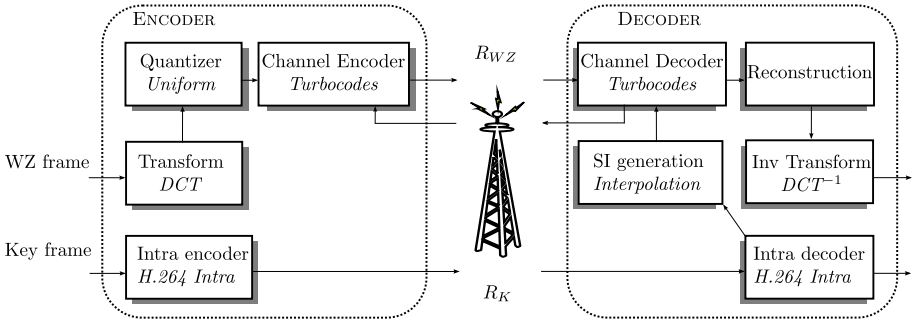


Figure 1: Schéma de codage vidéo distribué inspiré de Stanford, basé DCT.

Un domaine nouveau est également apparu ces dernières années, c'est celui des systèmes multicaméra (vision stéréo, vidéo 3D, ...). Dans ce genre de système de codage, le problème de la complexité des codeurs actuels se pose de la même façon. À cela, s'ajoute le fait que l'extraction de la corrélation entre images, comme elle se fait usuellement, nécessite une connaissance des images des caméras voisines, et donc nécessite tout ce qui en découle en matière d'installation, de communication, *etc.* Le codage vidéo distribué présente donc un double avantage s'il est appliqué aux systèmes multivues : celui de réduire la complexité et celui de supprimer les communications entre caméras difficiles à mettre en oeuvre. Le schéma du codage vidéo distribué multivue (CVDM) est similaire à celui du CVD monovue, apportant néanmoins de nouvelles problématiques. Les premières applications du CVDM peuvent se retrouver dans [Guo *et al.*, 2006a; Artigas *et al.*, 2006; Ouaret *et al.*, 2006].

La thèse synthétisée ici traite donc du CVDM fondé sur un codeur adoptant la structure de Stanford. Nous détaillons dans le Chapitre 1 de la thèse, les différents blocs de ce schéma et les différentes techniques proposées dans la littérature. A chaque fois, nous en dégageons des problématiques que nous nous proposons de traiter dans la suite du manuscrit.

Modèle débit-distorsion et son application

Résumé des chapitre 2 et 3 du manuscrit de thèse.

Modèle débit-distorsion

Les performances générales du codeur vidéo distribué dépendent en partie de la qualité de l'information adjacente. En effet, plus l'information adjacente est différente de la trame WZ initiale, plus le débit requis par le turbodécodeur est grand. Nous nous proposons donc, dans le Chapitre 2, d'établir une expression pour modéliser la variance de cette

erreur au décodeur. Nous obtenons l'expression suivante :

$$\hat{\sigma}^2_{e_I} = M_{d_1,d_2} + k_1^2 D_{I_1} + k_2^2 D_{I_2}.$$

où $\hat{\sigma}^2_{e_I}$ est la variance de l'erreur que l'on cherche à estimer. Le terme M_{d_1,d_2} correspond à l'erreur d'estimation de l'interpolation dans le cas où celle-ci est générée en utilisant des trames de référence non quantifiées (alors qu'en pratique elles sont quantifiées). Les termes D_{I_1} et D_{I_2} correspondent aux erreurs de quantification des images de référence et les coefficients k_1 et k_2 dépendent des distances entre les trames de références et la trame WZ estimée.

Dans ce chapitre nous proposons également un certains nombre de tests afin de valider notre modèle. Pour cela, nous considérons les différentes hypothèses nécessaire à l'obtention de la formule ci-dessus. Malgré des imprécisions à bas-débit, il résulte de ces tests que le modèle permet une estimation très acceptable de la distorsion observée en pratique.

L'avantage de notre modèle est sans nul doute dans la simplicité de son expression. En effet, les différents facteurs impactant sur la distorsion finale sont séparés en termes indépendants : d'un côté le terme M_{d_1,d_2} mesure l'erreur provenant de l'activité de mouvement dans la séquence. C'est donc une erreur intrinsèque dépendant uniquement du contenu de la vidéo. Au contraire, les distorsions D_{I_1} et D_{I_2} sont dues uniquement à la quantification et donc au choix extérieur du compromis débit-distorsion. Cette structure simple nous permet plus aisément de modéliser le comportement général du codeur, et nous proposons dans le chapitre suivant, d'utiliser ce modèle pour comprendre et optimiser le codeurs.

Etude des schémas multi-vues

Le codage vidéo distribué multivue, bien qu'il soit fondé sur la même stratégie de codage que le CVD monovue, apporte de nouvelles possibilités et avec elles, de nouveaux problèmes. L'apport le plus remarquable est celui concernant la génération d'information adjacente. Dans les schémas CVD utilisés pour ces travaux, l'estimation de la TWZ au décodeur est construite grâce à une méthode d'interpolation entre deux trames. Plus de détails seront donnés dans la partie dédiée à la construction de l'IA, mais ce qu'il faut retenir est que les méthodes usuelles effectuent une interpolation d'image grâce à deux TC encadrant la TWZ à estimer. Dans le codage monovue, il n'y a qu'un sens d'interpolation (le sens temporel). L'aspect multicaméra permet une interpolation fondée sur des TC n'appartenant pas à la même vue. Cela permet de construire une estimation de meilleure qualité, mais cela apporte de nouvelles questions concernant la position des trames dans le plan bidimensionnel "temps-vues". Dans les figures 2 et 3, nous donnons deux exemples de schémas existants dans la littérature. Il est évident, en vue de ces deux figures que les stratégies de décodage pour les deux schémas présentés seront totalement différentes. En effet, pour le schéma asymétrique, figure 3, l'IA ne pourra être générée que dans le sens des vues, alors que pour le schéma symétrique $\frac{1}{2}$ une interpolation temporelle et une interpolation intervue seront disponibles pour générer l'IA finale qui sera alors turbodécodée. L'étape intermédiaire, qui passe de deux interpolations à une IA unique est appelée *fusion* et est détaillée plus loin dans le document. En dehors de toute considération débit-distorsion, le choix du schéma a des conséquences sur les techniques mises en oeuvres pour

le codage (interpolation, fusion, estimation de paramètres...), mais également sur le choix du matériel de capture vidéo. En effet, si une caméra encode des TC, elle aura besoin d'une puissance de calcul plus importante que si celle-ci encode simplement des TWZ.

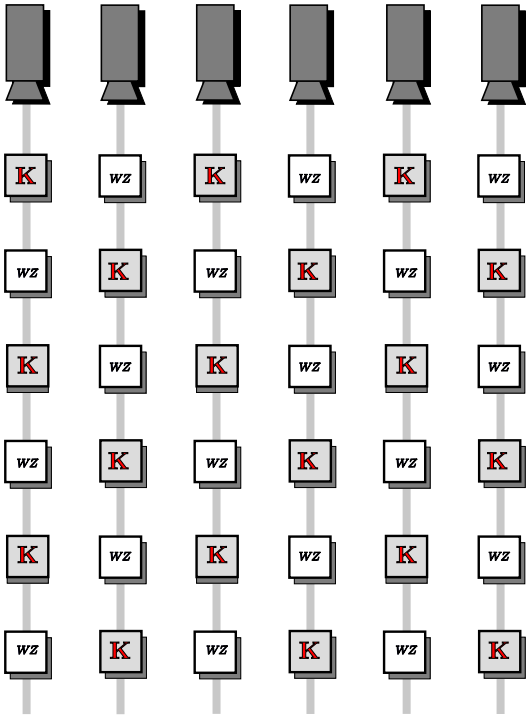


Figure 2: Schéma symétrique $\frac{1}{2}$

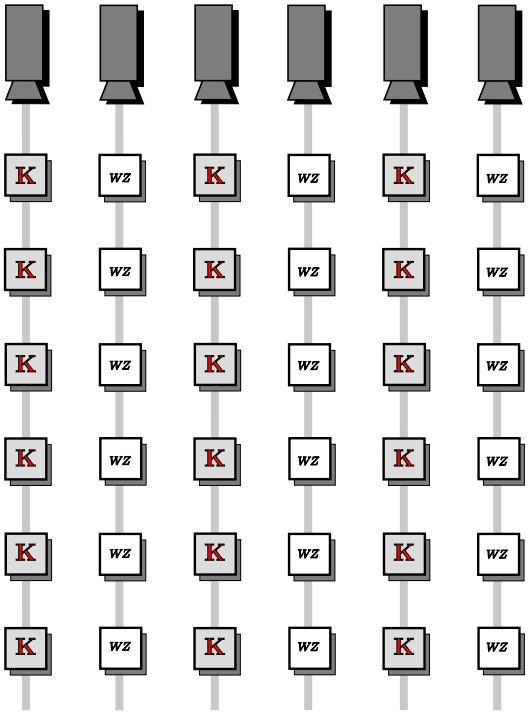


Figure 3: Schéma asymétrique

Dans le Chapitre 3, nous avons établi un état de l'art des schémas de codage existant dans la littérature. Les conclusions que nous en avons tirées nous amènent à penser que les schémas existants comportent un nombre trop élevés de TC, ce qui implique une complexité d'encodage encore trop forte, et des résultats débit-distorsion sous-optimaux. C'est pourquoi nous proposons également dans ce chapitre un nouveau schéma symétrique comportant moins de TC et donc allégeant l'encodage tout en améliorant les performances de codage.vEnfin, nous avons utilisé le modèle débit-distorsion proposé plus tôt afin d'étudier différentes stratégies de décodage envisageables dans ce nouveau schéma et nous avons pu déterminer la meilleure d'entre elles. La répartition des trames et l'ordre de décodage choisi peuvent être observés dans la figure 4. Enfin, les résultats débit-distorsion de la figure 5 montrent que le schéma proposé est plus performant que ceux existants déjà (en plus que d'être moins complexe et donc plus adapté à l'esprit distribué).

Etude de la propagation d'erreur en cas de perte de trame

Les schémas monovues ne présentent pas la même latitude qu'en multivue en ce qui concerne la disposition des types de trames. En effet, le seul paramètre modifiable est la taille des groupe d'images (Group of Pictures, GOP). Celle-ci est très souvent fixe, mais il existe des algorithmes où la taille est adaptative [Ascenso *et al.*, 2006]. Pour une taille de GOP fixée, il y a en revanche plusieurs stratégies de décodage possibles, *i.e.*, l'ordre

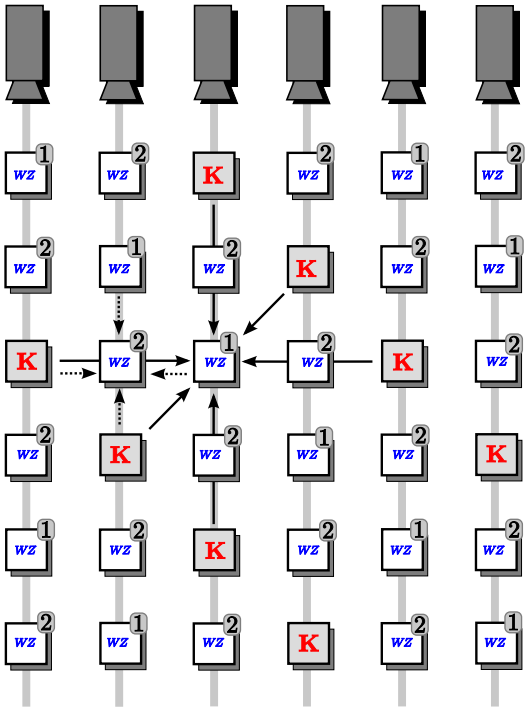


Figure 4: Schéma symétrique $\frac{1}{4}$ proposé dans la thèse

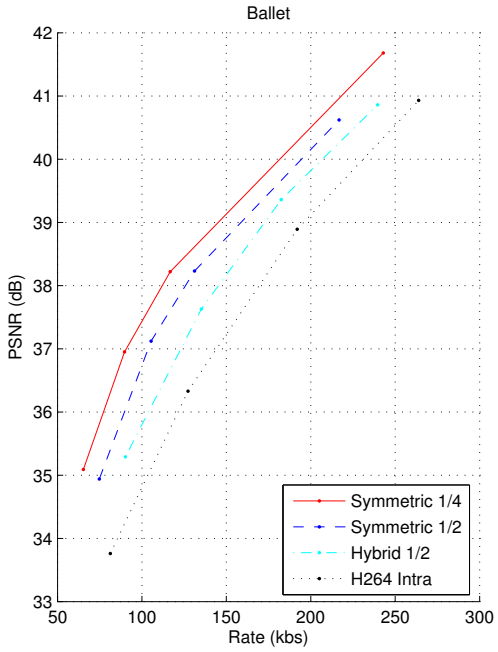


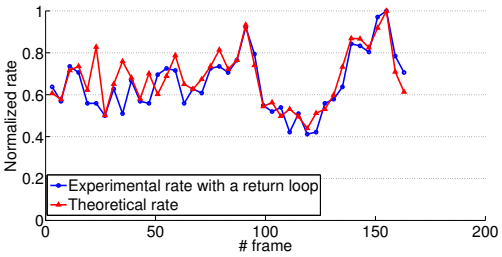
Figure 5: Résultats débit-distorsion du schéma proposé

de décodage des TWZ peut varier. Dans le Chapitre 3, nous avons déterminé, grâce au modèle débit-distorsion proposé, la meilleure des stratégies pour un cas de GOP 4. Un élément à prendre en compte également dans le choix d'un schéma de codage, est celui du phénomène de propagation d'erreur. Les différentes stratégies de décodage ne sont pas sensibles de la même manière aux pertes de trames dans le GOP. Dans le Chapitre 3, nous avons étudié le phénomène de propagation d'erreur dans un schéma monovue de taille 4. Grâce au modèle débit-distorsion, nous avons pu anticiper le comportement du codeur en cas de perte de trames lors de la transmission. Cela peut s'avérer utile lors du choix de la stratégie de codage, ou bien lors de l'allocation de débit à l'encodeur.

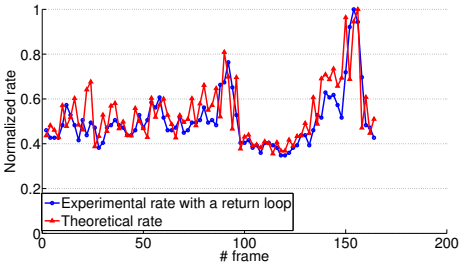
Contrôle du débit à l'encodeur permettant de s'affranchir de la boucle de retour

Une des problèmes liés au schéma de codage de DISCOVER est la limitation de la boucle de retour. En effet, comme il n'existe aucune méthode permettant d'estimer à l'encodeur le nombre de bits de parité à envoyer au turbodécodeur afin de permettre une reconstruction acceptable, les schémas actuels nécessitent l'utilisation d'une boucle de retour. Le décodeur reçoit une première salve de bits de parité, puis estime la probabilité d'erreur dans le signal reconstruit. Si celle-ci est trop grande (comparée généralement à un seuil), alors le décodeur demande des bits supplémentaires, par l'intermédiaire de ce canal de retour. L'utilisation de ce canal est évidement très peu envisageable dans une implantation pratique du schéma.

Dans le Chapitre 3, nous proposons un algorithme d'estimation de débit à l'encodeur qui permet de supprimer ce canal de retour. Voici une brève description du principe de



(a) W_m de la séquence *foreman* (CIF)



(b) W_l de la séquence *foreman* (CIF)

Figure 6: Comparaison entre le débits estimés et obtenus avec boucle de retour (les valeurs sont normalisées).

notre approche développée, sans perte de généralité, dans le cas d'une configuration où la taille des groupes d'image est fixée à 4.

Dans un premier temps, en se fondant sur le modèle proposé plus haut, nous déterminons la valeur des débits par trame :

$$R_m = \frac{1}{2}\log_2\left(\frac{\mu_m\left(M_{2,2}+\frac{1}{2}D_K\right)}{D_K}\right)$$
$$R_l = \frac{1}{2}\log_2\left(\frac{\mu_l\left(M_{1,1}+\frac{1}{2}D_K\right)}{D_K}\right).$$

où R_m et R_l sont respectivement les débits des trames du milieu du groupe d'image et des trames latérales (les deux autours de la trame du milieu). D_K correspond à la distorsion de la trame clef précédente. Ces expression ont été obtenues en fixant une contrainte qui force les distorsions des trames à être constantes le long de la séquence (une contrainte fortement liée au confort visuel). On peut voir des les figures 6 (a) et (b) que les débits estimés correspondent bien aux débits idéaux obtenus avec une boucle de retour.

Dans un second temps, l'algorithme partage le débit estimé juste avant entre les différents plans de bits des différentes sous-bandes. Ce débit par sous-bande est estimé en s'appuyant sur la distance de Hamming entre la WZ originale et une estimation (très simple) de l'information adjacente. Cette technique présente des résultats débit-distorsions acceptables dans lesquels notre schéma dégrade de seulement 0.6 dB les résultats obtenus dans le cas idéal, c'est à dire celui avec utilisation de la boucle de retour.

Génération de l'information adjacente

Résumé des chapitre 4, 5, 6 et 7 du manuscrit de thèse.

Méthodes de référence

Dans cette partie nous présenterons plus précisément tout ce qui concerne les méthodes d'interpolation utilisées pour générer l'information adjacente. Notons que certaines solutions proposent d'utiliser d'autres approches type extrapolation [Natario *et al.*, 2005] pour se détacher des problèmes liés aux interpolations. Pourtant cette famille de méthodes restent les plus performantes aujourd'hui, et ces pourquoi nous nous concentrerons sur ce

type d'approches. Comme indiqué dans la figure 7, pour estimer une TWZ, les algorithmes d'interpolation nécessitent deux TC encadrant la TWZ. Celles-ci ne doivent pas forcément être les trames immédiatement voisines de la TWZ, elles peuvent se situer plus loin dans la vidéo. Les algorithmes d'interpolation ont pour but d'estimer les deux champs de vecteurs reliant la TWZ à chacune des TC. Ceux-ci sont utilisés pour compenser les TC et construire alors une estimation de la TWZ.

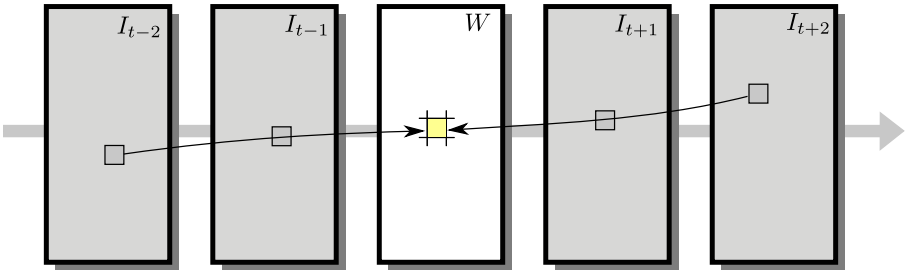


Figure 7: Interpolation de la TWZ entre deux TC. Les champs de vecteurs estimés sont utilisés pour la compensation qui consiste à moyenner les deux blocs des TC.

Dans le cadre du projet européen DISCOVER, un algorithme [Ascenso *et al.*, 2005a] en quatre étapes qui s'avère être l'un des plus efficaces parmi les méthodes existantes, a été proposé. La première étape consiste à filtrer les TC afin d'augmenter la robustesse de la méthode. Ensuite un premier champ de vecteur est calculé entre les deux TC (utilisant un algorithme de recherche par bloc). Ce champ de vecteur sert de base à la construction d'un champ bidirectionnel cette fois-ci entre la TWZ et les deux TC. Une troisième étape consiste à raffiner ce champ bidirectionnel, à nouveau à l'aide d'un algorithme de recherche par bloc. La dernière étape est une opération de filtrage (filtrage médian) sur les vecteurs obtenus.

Cette méthode, très efficace, a souvent été utilisée pour effectuer des interpolations intervues [Areia *et al.*, 2007]. Toutefois, si dans le domaine temporel elle permet une très bonne estimation des mouvements de la scène, dans le sens des vues, sa structure est limitée et ne délivre pas une bonne appréciation de la structure de la scène, nécessaire pour une interpolation. Cependant, même si ces résultats sont moins bons que pour une interpolation de mouvement, elle donne de meilleurs résultats que beaucoup de méthodes existantes dans le sens des vues.

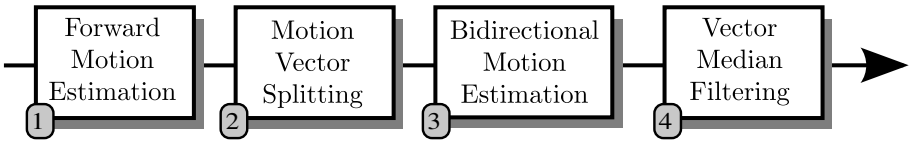


Figure 8: Schéma général des méthodes d'interpolation denses proposées. Les blocs en traits pleins constituent les étapes de l'algorithme de DISCOVER, et ceux en traits pointillés, constituent les raffinements de champs de vecteurs.

Interpolation développée dans le cadre du projet Essor

Dans le cadre du projet ESSOR, nous avons proposé une nouvelle méthode d'interpolation par bloc qui obtient de meilleures performances que Discover. Celle-ci effectue deux estimations de champs de vecteurs monodirectionnels entre les TC (une dans chaque sens), ensuite selon une méthode prenant en compte les pixels indépendamment, elle construit une estimation qui s'avère être bien souvent meilleure que celle générée avec Discover. Cette méthode propose donc une première solution pour se détacher de la description par bloc de Discover en considèrent les pixels un par un. Seulement, les champs de vecteurs sont encore par bloc. Dans la suite, nous proposons d'estimer directement les champs de vecteur pixel par pixel.

Méthodes denses

Interpolation denses

Toutes ces méthodes adoptent une approche par bloc, *i.e.*, elles utilisent un champ de vecteurs par bloc (généralement de taille 8×8 pixels). Cela est justifié dans les schémas de codage classiques type H.264, car les estimations de mouvement se font à l'encodeur et les champs obtenus sont alors transmis donc économisés. En revanche, dans des schémas de CVD, ces étapes d'interpolation sont effectuées au décodeur, et il n'y a aucune raison de limiter le nombre de vecteurs, sauf bien sûr pour des raisons de complexité mais l'hypothèse est souvent faite en CVD que la complexité au décodeur n'est pas un problème. Ainsi, nous proposons une famille de méthodes effectuant une interpolation dense, *i.e.*,un vecteur par pixel (figure 8).

Estimer un champ dense n'est pas un problème si simple car en augmentant le nombre de vecteurs, on diminue inévitablement la stabilité. Ainsi, nous proposons d'utiliser deux techniques de raffinement, permettant de rendre dense un champ initialement décrit par bloc. Les deux techniques de raffinement de champs de vecteurs utilisées sont celles reposant sur l'algorithme de Cafforio-Rocca et sur une approche variationnelle.

Le descriptif détaillé des méthodes est donné dans le Chapitre 6. Pour résumer, le premier algorithme de raffinement propose pixel par pixel une correction optimale d'une valeur initiale bien choisie en fonction des voisins. La deuxième adopte plutôt une approche variationnelle ayant pour but d'obtenir un champ lisse dans l'ensemble mais avec changement brutal au niveau des contours. On peut voir dans le tableau 1 que pour certaines séquences, le gain par rapport à la méthode de référence Discover est très fort. En revanche, le gain est plus faible pour d'autres. Cela est du au fait que les performances obtenues dépendent encore fortement des paramètres internes aux méthodes. Cela étant, les résultats restent encourageants et nous invite à trouver un moyen d'adapter ces paramètres aux contenu des séquences.

Méthodes de fusion

Dans la section précédente, nous avons présenté les méthodes d'interpolation d'images utilisées. Dans les schémas où il y a une interpolation par TWZ, l'estimation résultante constitue l'IA à turbodécoder. En revanche, dans la plupart des schémas multivues, au moins deux interpolations sont effectuées (temps et vues), et ainsi deux estimations doivent être fusionnées afin de constituer une unique IA. Dans cette section, nous présentons, et

	CD	DC	VD	Mean
akiyo*	0.00	-0.08	0.00	-0.02
city*	0.93	0.17	1.12	0.74
container*	0.17	-0.23	0.17	0.04
eric*	0.18	-0.16	0.24	0.08
football*	-0.27	-0.12	-0.16	-0.18
foreman*	0.20	0.21	0.20	0.21
mother and daughter*	0.01	0.00	0.01	0.01
mobile*	0.84	0.00	1.03	0.62
news*	0.09	0.00	0.09	0.06
tempete*	-0.10	-0.01	-0.08	-0.06
silent*	-0.02	0.02	-0.02	-0.01
waterfall*	0.01	0.01	0.01	0.01
planet* (synthetic sequence)	0.09	0.22	0.14	0.15
book arrival$^+$	-0.12	0.07	-0.11	-0.05
outdoor$^+$	0.25	0.03	0.29	0.19
ballet$^+$	0.13	0.04	0.15	0.11
ballroom$^+$	-0.04	0.06	0.01	0.01
uli$^+$	-0.00	0.03	0.02	0.02
Moyenne	**0.13**	**0.02**	**0.17**	**0.11**

Table 1: Δ PSNR moyen sur les IA dans le sens temporel et pour plusieurs séquences. *: séquences monovues (352×288, 30 fps), $^+$: séquences multivues (512×384, 30 fps)

nous mettrons en équation les fusions existantes et nous proposerons trois nouvelles fusions. Cette section est tirée de travaux présentés dans [Maugey *et al.*, 2009].

La figure 9 représente les différents éléments rentrant en jeu lors de la fusion. L'hypothèse est qu'il faut créer une unique IA pour le décodage d'une TWZ nommée, $W_{n,t}$. Pour cela quatre TC sont disponibles: $I_{n,t\text{-}1}$, $I_{n,t+1}$, $I_{n\text{-}1,t}$ et $I_{n+1,t}$. Les interpolations ont donnés quatre champs de vecteurs, $\mathbf{v}_b$, $\mathbf{v}_f$, $\mathbf{v}_l$ et $\mathbf{v}_r$ afin de compenser ces TC et donner quatre estimations $\tilde{I}_{n,t^-}$, $\tilde{I}_{n,t^+}$, $\tilde{I}_{n^-,t}$, $\tilde{I}_{n^+,t}$. Les méthodes considèrent souvent qu'il n'y a que deux estimations, car les deux temporelles ainsi que les deux intervues sont souvent moyennées pour ne générer qu'une seule estimation temporelle et une seule intervue.

Méthodes existantes

Avec les notations de la figure 9, voici une liste des fusions les plus performantes existantes. Les fusions existantes sont dites "binaires", c'est-à-dire que pixel par pixel elles choisissent la meilleure valeur parmi celles disponibles. Par la suite, nous les opposerons aux fusions "linéaires" qui effectuent une combinaison des valeurs disponibles.

La **fusion idéale** (Id) étudiée dans [Areia *et al.*, 2007; Maugey, Pesquet-Popescu, 2008], correspond à la borne supérieure des fusions binaires. Pixel par pixel, la meilleure des estimations est choisie en calculant la véritable erreur:

$$\tilde{I}(\mathbf{s}) = \begin{cases} \tilde{I}_N(\mathbf{s}), & \text{if } |\tilde{I}_N(\mathbf{s}) - W_{n,t}(\mathbf{s})| < |\tilde{I}_T(\mathbf{s}) - W_{n,t}(\mathbf{s})| \\ \tilde{I}_T(\mathbf{s}), & \text{sinon.} \end{cases}$$

La **fusion par différence de pixels** (PD), proposée par Ouaret *et al.* [Ouaret *et al.*,

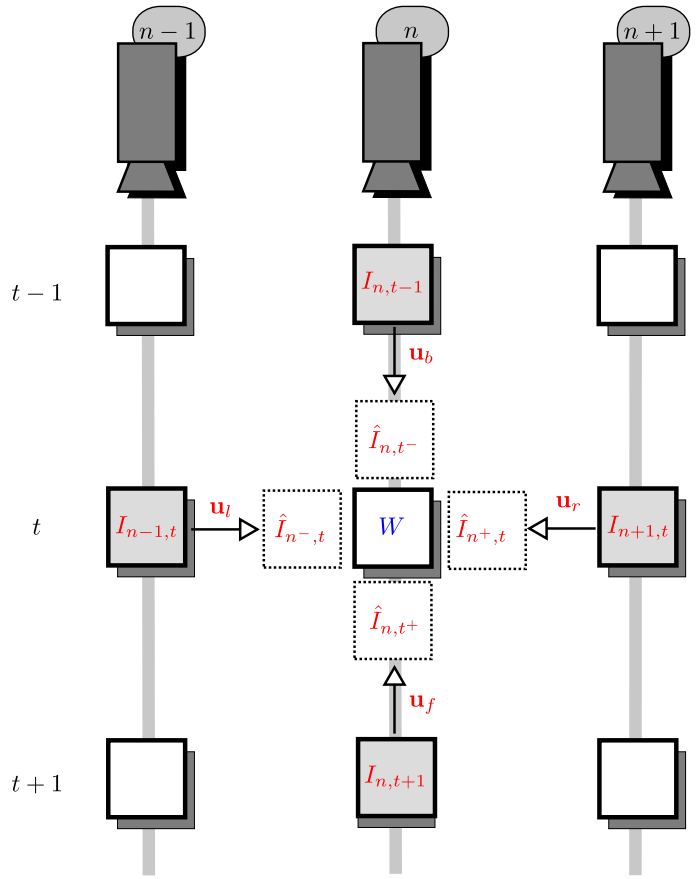


Figure 9: Problématique de la fusion de quatre estimations pour une TWZ à un instant t de la caméra n: les I_x correspondent aux TC disponibles au décodeur et $\tilde{I}_x$ à leurs versions compensées en mouvement, estimant $W_{n,t}$. Les $\mathbf{v}_x$ correspondent aux vecteurs de mouvement.

2006] dans laquelle l'erreur d'estimation est approximée grâce aux trames placées avant et après la TWZ courante:

$$\tilde{I}(\mathbf{s}) = \begin{cases} \tilde{I}_N(\mathbf{s}), & \text{si } E_N^b(\mathbf{s}) < E_T^b(\mathbf{s}) \text{ et } E_N^f(\mathbf{s}) < E_T^f(\mathbf{s}) \\ \tilde{I}_T(\mathbf{s}), & \text{sinon.} \end{cases}$$

où $E_N^b = |\tilde{I}_N - I_{n,t-1}|$, $E_N^f = |\tilde{I}_N - I_{n,t+1}|$, $E_T^b = |\tilde{I}_T - I_{n,t-1}|$ et $E_T^f = |\tilde{I}_T - I_{n,t+1}|$.

La **fusion par différence des compensations en mouvement** (MCD) proposée dans [Guo *et al.*, 2006a] et dans laquelle la valeur absolue de la différence entre les deux TC temporelles compensées est seuillée ainsi que la valeur des vecteurs de mouvement:

$$\tilde{I}(\mathbf{s}) = \begin{cases} \tilde{I}_N(\mathbf{s}), & \text{si } |\tilde{I}_{n,t^-}(\mathbf{s}) - \tilde{I}_{n,t^+}(\mathbf{s})| > T_1 \\ & \quad \text{ou } \|\mathbf{v}_b(\mathbf{s})\| > T_2 \\ & \quad \text{ou } \|\mathbf{v}_f(\mathbf{s})\| > T_2 \\ \tilde{I}_T(\mathbf{s}), & \text{sinon.} \end{cases}$$

La **fusion par projection selon les vues** (Vproj) [Ferre *et al.*, 2007] consiste à projeter l'estimation temporelle sur les vues adjacentes ($dc_l(\cdot)$ et $dc_r(\cdot)$). On calcule ensuite

les deux erreurs avec les TC des vues adjacentes (E_l et E_r). Celles-ci sont à nouveau compensées en disparité vers la vue centrale:

$$\tilde{I}(\mathbf{s}) = \begin{cases} \tilde{I}_N(\mathbf{s}), & \text{si } |dc_l^{-1}(E_l)(\mathbf{s})| > T \text{ ou } |dc_r^{-1}(E_r)(\mathbf{s})| > T \\ \tilde{I}_T(\mathbf{s}), & \text{sinon.} \end{cases}$$

La **fusion par projection temporelle** (Tproj) [Ferre *et al.*, 2007] qui est la version temporelle de la Vproj.

$$\tilde{I}(\mathbf{s}) = \begin{cases} \tilde{I}_N(\mathbf{s}), & \text{si } mc_b^{-1}(E_b) < T \text{ ou } mc_f^{-1}(E_f) < T \\ \tilde{I}_T(\mathbf{s}), & \text{sinon.} \end{cases}$$

Méthodes proposées

La première des méthodes proposées, comme celles de l'état de l'art est dite "binaire":
La **fusion binaire par différence des compensations en mouvement et disparité** (MDCDBin) compare les résidus des estimations temporelles et intervues qui sont définis par $E_T(\mathbf{s}) = |\tilde{I}_{n,t^-}(\mathbf{s}) - \tilde{I}_{n,t^+}(\mathbf{s})|$ et $E_N(\mathbf{s}) = |\tilde{I}_{n^-,t}(\mathbf{s}) - \tilde{I}_{n^+,t}(\mathbf{s})|$.

$$\tilde{I}(\mathbf{s}) = \begin{cases} \tilde{I}_N(\mathbf{s}), & \text{si } E_N(\mathbf{s}) < E_T(\mathbf{s}) \\ \tilde{I}_T(\mathbf{s}), & \text{sinon.} \end{cases}$$

L'approche innovante de nos travaux est de proposer et de tester deux fusions dites linéaires pour lesquelles la valeur de l'estimation finale est une combinaison linéaire des estimations disponibles. Les coefficients de cette combinaison sont déterminés en fonction de plusieurs paramètres.
Dans la **fusion linéaire par différence des compensations en mouvement et disparité** (MDCDLin), les résidus E_T and E_N sont utilisés pour bâtir les coefficients:

$$\tilde{I}(\mathbf{s}) = \frac{E_T(\mathbf{s})}{E_T(\mathbf{s}) + E_N(\mathbf{s})}\tilde{I}_N(\mathbf{s}) + \frac{E_N(\mathbf{s})}{E_T(\mathbf{s}) + E_N(\mathbf{s})}\tilde{I}_T(\mathbf{s})$$

L'idée de la **fusion linéaire fondée sur l'erreur d'estimation et la norme des vecteurs** (ErrNorm), est de prendre en compte l'information concernant la taille des vecteurs de mouvement:

$$\begin{aligned} \tilde{I}(\mathbf{s}) &= \frac{\tilde{I}_{\text{err}}(\mathbf{s}) + \tilde{I}_{\text{norm}}(\mathbf{s})}{2} \qquad \text{où} \\ \tilde{I}_{\text{norm}}(\mathbf{s}) &= \frac{(\|\mathbf{v}_b\| + \|\mathbf{v}_f\|)\tilde{I}_N(\mathbf{s}) + (\|\mathbf{v}_l\| + \|\mathbf{v}_r\|)\tilde{I}_T(\mathbf{s})}{\|\mathbf{v}_b\| + \|\mathbf{v}_f\| + \|\mathbf{v}_l\| + \|\mathbf{v}_r\|} \\ \tilde{I}_{\text{err}}(\mathbf{s}) &= \frac{E_T(\mathbf{s})\tilde{I}_N(\mathbf{s})}{E_T(\mathbf{s}) + E_N(\mathbf{s})} + \frac{E_N(\mathbf{s})\tilde{I}_T(\mathbf{s})}{E_T(\mathbf{s}) + E_N(\mathbf{s})} \end{aligned}$$

Les résultats expérimentaux (figure 10) pour ces méthodes proposées sont encourageants car elles obtiennent de meilleures performances débit-distorsion que les méthodes existantes.

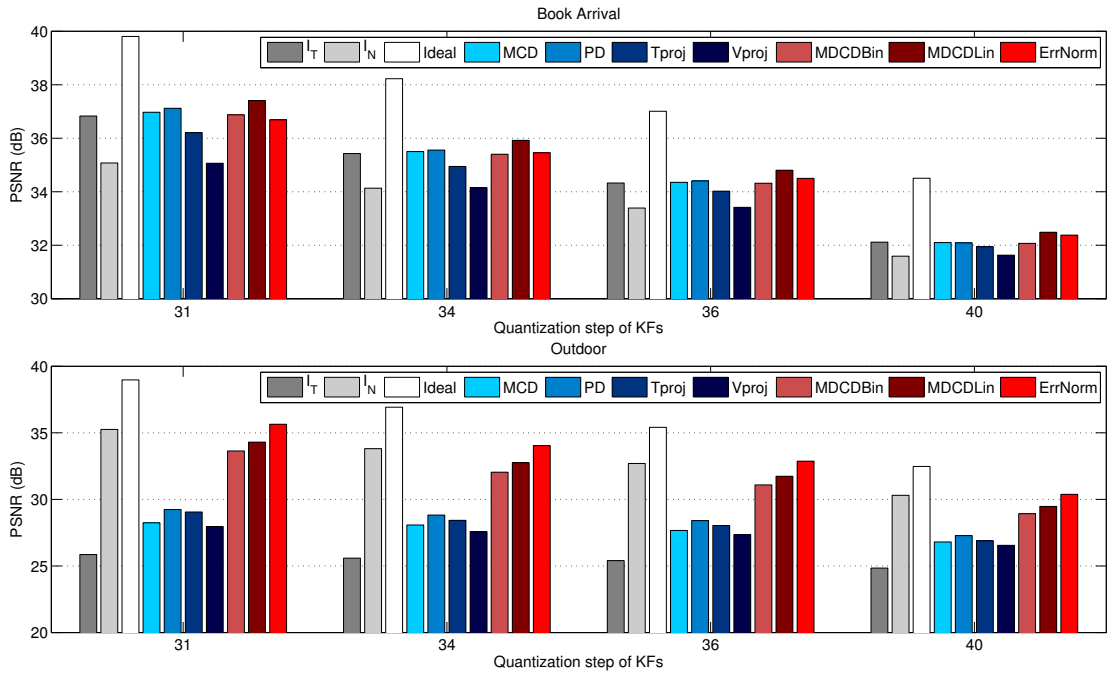


Figure 10: Qualité de l'IA pour différentes méthodes de fusion, pour différent pas de quantification des TC, et pour deux séquences *book arrival* and *outdoor*.

Schémas à base d'information de hachage

Partant du principe qu'il existe de zones dans l'image qui ne peuvent être estimées au décodeur (car ces zones ne sont pas présentes dans les TC, comme dans le cas d'occlusions par exemple), certaines schémas envoient une petite partie de l'information WZ afin d'aider l'estimation de l'IA et du bruit de corrélation dans ces zones. Ces schémas, dits à base d'information de hachage, soulèvent plusieurs problématiques: le choix de l'information de hachage à transmettre, son mode de compression, et son utilisation au décodeur.

Dans cette thèse nous proposons un nouveau schéma de ce type représenté dans la figure 11. Contrairement aux méthodes existant dans la littérature, nous avons choisi d'effectuer cette sélection au décodeur et d'utiliser le canal de retour pour transmettre à l'encodeur la sélection. Ainsi, au lieu d'avoir une mauvaise estimation de l'IA (typiquement une moyenne) mais la trame originale disponible comme les méthodes existantes l'ont, nous avons choisi de nous affranchir de la connaissance de la trame originale mais d'avoir à disposition la vraie IA estimée. Une fois la sélection faite, l'encodeur compresse ces informations de hachage retenues. Pour cela nous avons choisi d'utiliser la même matrice de quantification que celle adoptée dans Discover. Le choix des paramètres de quantification s'est effectué expérimentalement.

Une fois l'information de hachage sélectionnée, compressée et transmise, le décodeur utilise cette information supplémentaire pour construire une information adjacente plus précise. Celle-ci est obtenue grâce à une algorithme génétique où différents candidats sont fusionnés et sélectionnés selon les règles de l'évolution des êtres vivants. Les résultats obtenus par ce schéma sont présentés dans la figure 12. Ceux-ci montrent le potentiel de

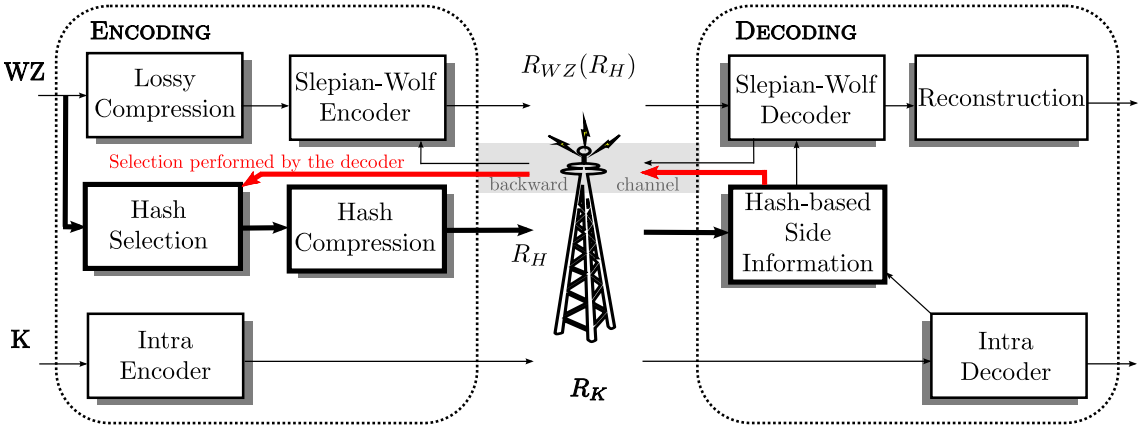


Figure 11: Structure générale du schéma à base d'information de hachage proposé. En rouge, le canal de retour qui constitue la spécificité de notre approche.

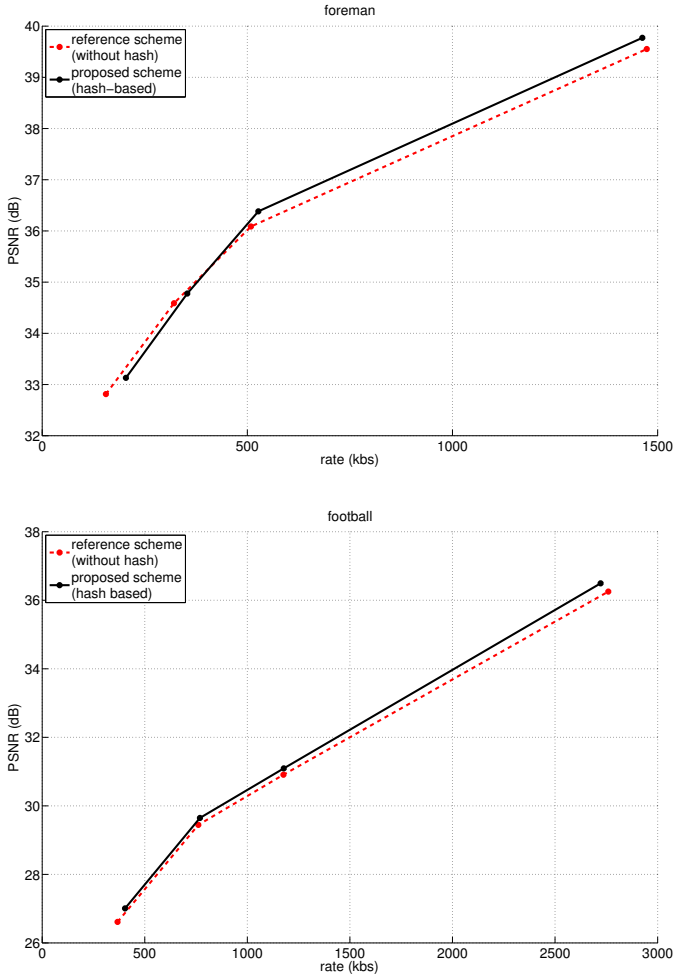


Figure 12: Performances débit-distorsion pour deux séquences au format CIF. En pointillés rouges, les performance du codeur de référence DISCOVER et en lignes noires et pleines l'algorithme proposé.

notre schéma et du fait de transmettre de l'information de hachage pour affiner l'estimation de l'information adjacente au décodeur.

Estimation du bruit de corrélation

Résumé du chapitre 8 du manuscrit de thèse.

Nous rappelons que dans la structure générale des codeurs CDV type Stanford, le décodeur corrige les IA avec les bits de parité envoyés par le codeur WZ. Ce processus est effectué sous l'hypothèse que l'erreur d'estimation peut être considérée comme une erreur de canal. Pour fonctionner, le turbo décodeur a besoin d'un modèle pour le bruit de corrélation entre la TWZ et son IA associée et les performances du codec dépendent en partie de la qualité de ce modèle. Le bruit de corrélation est généralement estimé à l'aide d'un modèle Laplacien [Aaron *et al.*, 2002].
Nous proposons dans cet article de remplacer le modèle Laplacien par un modèle Gaussien Généralisé (GG) recouvrant une large classe de distributions classiques comme les Gaussiennes ou les Laplaciennes. Il a été montré que ce modèle est bien adapté pour la représentation des coefficients d'ondelettes de signaux ou d'images [Mallat, 1989]. Notons enfin qu'il a été montré que cette distribution offre un bon modèle pour les coefficients de DCT d'images naturelles [Müller, 1993]. Ces propriétés peuvent conduire à appliquer ce modèle aux deux transformations usuelles en compression d'images et de vidéos que sont la DCT et les ondelettes.

Soit X la TWZ originale et soient I_p et I_s les trames de références construites à partir des TC précédente et suivante. Au décodeur, l'IA est notée Y et le résidu R est défini comme la différence entre les trames I_p et I_s compensées. Soit $\mathbf{s} = (x, y)$ un pixel et en notant les champs de vecteurs de mouvement précédent et suivant par MV_p et MV_s, alors Y et R s'expriment de la manière suivante :

$$Y(\mathbf{s}) = \frac{I_p(\mathbf{s} + MV_p(\mathbf{s})) + I_s(\mathbf{s} + MV_s(\mathbf{s}))}{2}, \tag{1}$$

$$R(\mathbf{s}) = \frac{I_p(\mathbf{s} + MV_p(\mathbf{s})) - I_s(\mathbf{s} + MV_s(\mathbf{s}))}{2}. \tag{2}$$

X, Y et R peuvent être transformés à l'aide d'une DCT entière 4×4 ou à l'aide d'une transformée en ondelette biorthogonale de type 9/7 (sur 3 niveaux de décomposition). Nous notons ainsi par $x_{k,i}$, $y_{k,i}$ et $r_{k,i}$ les $i^{\text{ème}}$ coefficients de la $k^{\text{ème}}$ sous-bande ($k \in [1, ..., K]$ et $i \in [1, ..., N_k]$), résultant de la décomposition de X, Y et R.

Une hypothèse classique en CVD est de considérer que la corrélation dépend uniquement de la sous-bande et que le bruit est modélisé par une distribution Laplacienne. Dans les premiers travaux sur le CVD [Aaron *et al.*, 2002], l'estimation des coefficients se faisait hors-ligne. Autrement dit, il était supposé que les paramètres $(\alpha_k)_{k=1}^{K}$ de chaque sous-bande sont connus par le décodeur. Cette hypothèse, peu réaliste puisqu'elle suppose connue l'erreur $x_{k,i} - y_{k,i}$ au décodeur, a ensuite été remplacée par une solution en-ligne [Brites, Pereira, 2008] qui consiste à estimer l'erreur à l'aide des coefficients du résidu $r_{k,i}$.

La densité de probabilité d'une GG de moyenne nulle et de paramètres $(\alpha, \beta) \in \mathbb{R}_+^{*2}$:

$$f_{\alpha,\beta}(x) = \frac{\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)} e^{-\left(\frac{|x|}{\alpha}\right)^{\beta}},$$

où $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ est la fonction Gamma d'Euler (si $\beta = 1$ on retrouve la densité d'une Laplacienne). Nous proposons d'estimer les paramètres de la densité de probabilité de deux manières: méthodes des moments, et méthodes du maximum de vraisemblance.

Méthode 1	Méthode 2	*city*	*football*	*foreman*
Lap hors-ligne	GG hors-ligne MV	-0,96	-3,73	-1,78
Lap hors-ligne	GG hors-ligne Mom	1,21	-3,61	-1,52
Lap en-ligne	GG en-ligne MV	0,36	-3,29	-0,90
Lap en-ligne	GG en-ligne Mom	-1,30	-4,30	-1,88
Lap hors-ligne	Lap en-ligne	1,73	2,67	1,53
GG hors-ligne MV	GG en-ligne Mom	1,40	2,10	1,39
Lap hors-ligne	GG en-ligne Mom	0,44	-1,64	-0,38

Table 2: Gains en débit (%) de la méthode 2 par rapport à la méthode 1 sur différentes séquences.

Le tableau 2 montre un exemple de résultat que l'on obtient en changeant le modèle Laplacien par un modèle GG dans le cas d'une transformée DCT entière 4×4. Les gains en débit sont calculés à l'aide de la « métrique » de Bjontegaard [Bjontegaard, 2001]. On constate que sur toutes les séquences testées la méthode GG permet de diminuer le débit aussi bien en mode hors-ligne (jusqu'à $3,73\%$ sur Football CIF et $1,78\%$ sur Foreman QCIF) qu'en mode en-ligne. Pour un PSNR de $38,38$dB sur la séquence Football cela correspond à une réduction de 194kbs hors-ligne et 128kbs en-ligne, et sur Foreman à $39,94$dB les différences sont de 44kbs hors-ligne et 46kbs en-ligne. On peut noter que la méthode MV semble plus performante en hors-ligne et que la méthode des moments donne de meilleurs résultats en mode en-ligne. Finalement, on peut remarquer qu'en utilisant GG Mom en mode en-ligne on peut, sur certaines séquences, obtenir des gains par rapport aux résultats avec le modèle Laplacien hors-ligne.

Etude des métriques d'estimations de la qualité de l'information adjacente

Résumé du chapitre 9 du manuscrit de thèse.

Dans la littérature, la qualité de l'information adjacente est quasiment toujours mesurée grâce au PSNR :

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{EQM}\right)$$

où EQM est l'erreur quadratique moyenne entre l'image originale, I_{ref} et l'information

adjacente, I :

$$\mathrm{EQM} = \frac{1}{N_{\mathrm{width}} \times N_{\mathrm{height}}} \sum_{\mathbf{p} \in [\![1, N_{\mathrm{height}}]\!] \times [\![1, N_{\mathrm{width}}]\!]} \Big(I(\mathbf{p}) - I_{ref}(\mathbf{p}) \Big)^2 .$$

Bien que cette métrique soit celle la plus couramment adoptée, il existe des cas de figure où celle-ci ne parvient pas à bien indiquer la bonne valeur de la qualité (cf thèse de Denis Kubasov [Kubasov, 2008]). Dans la thèse nous avons fabriqué une autre situation dans laquelle le PSNR ne prédit pas le bon ordre de qualité entre deux information adjacentes. On peut voir dans la Figure 13 deux trames visant à estimer la même TWZ avec deux erreurs différentes. La première a été construite grâce à une interpolation classique et l'autre avec l'addition d'un bruit artificiel stationnaire à la trame originale. Ces deux estimations présentent le même PSNR (cf tableau 3). Or après décodage, il s'avère que l'interpolation obtient de bien meilleures performances que l'IA avec le bruit artificiel.

Type d'IA	PSNR de l'IA (dB)	débit (kb)	PSNR décodé (dB)
Interpolation de DISCOVER	29.05	137.28	39.29
Originale+ bruit artificiel	29.04	192.46	35.40

Table 3: Exemple des limites du PSNR comme métrique de qualité de l'IA.

(a)

(b)

Figure 13: Les deux IA du tableau 3 (a) DISCOVER 29.05 dB and (b) bruit artificiel 29.04 dB.

C'est au vue de ce constat que nous proposons de tester d'autres métriques qui viseraient à mesurer plus fidèlement la qualité de l'information adjacente. Une des métriques a été proposée par Kubasov dans sa thèse :

$$\mathrm{SIQ} = 10 \log_{10} \left(\frac{255^2}{\frac{1}{N_{\mathrm{width}} \times N_{\mathrm{height}}} \sum_{\mathbf{p} \in [\![1, N_{\mathrm{height}}]\!] \times [\![1, N_{\mathrm{width}}]\!]} \left| I(\mathbf{p}) - I_{ref}(\mathbf{p}) \right|^{\frac{1}{2}}} \right) . \tag{3}$$

Nous choisissons également d'étudier une version plus générale de cette mesure avec

$a > 0$:

$$\mathrm{SIQ}_a = 10\log_{10}\left(\frac{255^2}{\frac{1}{N_{\mathrm{width}}\times N_{\mathrm{height}}}\sum_{\mathbf{p}\in[\![1,N_{\mathrm{height}}]\!]\times[\![1,N_{\mathrm{width}}]\!]}\left|I(\mathbf{p})-I_{ref}(\mathbf{p})\right|^a}\right).$$

Dans le but d'élaborer une métrique la plus proche du fonctionnement du turbo décodeur, nous proposons la métrique suivant qui cumule les distances de Hamming sur tous les plans de bits et sur toutes les bandes :

$$\mathrm{HSIQ}(qi) = 10\log_{10}\left(\frac{1}{\frac{1}{N_{\mathrm{bits}}}\sum_b\sum_{bp}\sum_c \overline{I}(b,bp,c)\oplus\overline{I}_{ref}(b,bp,c)}\right) \tag{4}$$

où $\overline{I}$ et $\overline{I}_{ref}$ sont les versions transformées et quantifiées respectivement de l'IA et de la TWZ originale.

Pour tester ces métriques (PSNR, SIQ_a avec $a \leq 1$ et HSIQ), nous avons créé des bases de données d'IA pour lesquelles nous avons définie une «vraie» qualité fondée sur les résultats débit-distorsion après turbodécodage. Nous avons ensuite comparé cette «vraie» qualité aux qualité mesurées avec les métriques proposées. Voici les conclusions obtenues. Lorsqu'au sein d'une même base de données, il n'y a qu'un type d'erreur (erreurs d'estimation de mouvement par exemple), toutes les métriques, y compris le PSNR, sont fiables. Ce qui valide l'utilisation habituelle du PSNR. Seulement lorsque dans la base de données, plusieurs types d'erreur apparaissent (erreurs d'estimation de mouvement et de quantification des TCs), le PSNR obtient alors un score de fiabilité très faible, alors que les autres métriques demeurent performantes. C'est donc dans le cas où différents types d'erreur se côtoient que le PSNR présente ses limites. Cela est dû au fait que le PSNR tient compte plus fortement des grandes erreurs, alors que le turbodécodeur est sensible à tous types d'erreur (forte ou non) tout comme les SIQ_a proposées (avec $a \leq 1$) et la HSIQ.

Conclusion

Perspectives ou extension des travaux effectués

En se fondant sur les résultats et conclusions tirées de chacune de nos contributions, nous détaillons ici les différentes pistes qui seraient, selon nous, intéressantes d'explorer.

De nouveaux schémas multivues à base d'extrapolation contenant moins de trames clefs : le schéma symétrique proposé dans le Chapitre 3 obtenant de meilleurs résultats, il serait intéressant d'explorer encore plus de modes de classification, et les techniques de génération d'information adjacente qui en découleraient. Si l'utilisation d'interpolations limite en effet l'extension de la distance entre les trames clefs (car vraiment trop mauvaises pour des trames clefs trop éloignées), on pourrait songer à effectuer des extrapolations qui ne diminuent pas en performances quand l'éloignement des trames clefs s'accroit. Cela nécessiterait une élaboration de méthodes d'extrapolation multivues encore inexistantes aujourd'hui. En revanche, pour des schémas de ce type, une perte de trame peut s'avérer catastrophique pour les performances. Il serait donc intéressant

d'étudier ce phénomène grâce à une extension au multivue du modèle débit-distorsion proposé.

Un contrôle du débit étendu au multivue, moins dépendant de paramètres à estimer offline : une fois le modèle débit-distorsion étendu au multivue, il deviendra possible d'étendre pas la même l'algorithme de contrôle du débit proposé au cas de séquences multivue. En revanche, pour le cas monovue et multivue, il est également nécessaire de se pencher sur la mise en pratique de cette méthode. En effet, l'algorithme existant est trop dépendant de certains paramètrage effectué en amont et dépendant de la vidéo. Il faudrait ainsi pouvoir estimer ces coefficients en ligne, directement à l'encodeur.

Une meilleure adaptation en ligne des paramètres des méthodes d'estimations denses : les résultats obtenus dans le Chapitre 6 nous amènent au constat suivant : les méthodes proposées peuvent s'avérer très efficaces dans certaines situations, mais ne dépassent pas l'approche par bloc de DISCOVER dans d'autres cas de figure. Nous pensons que cela est dû à une trop forte dépendance de ces méthodes aux paramètres, et qu'il serait intéressant d'envisager une solution de détermination en ligne de ces paramètres.

Des méthodes de fusion fondées sur la reconnaissance de contour : après avoir exploré des fusions linéaires, il serait certainement profitable de fonder le calcul de coefficients de la combinaison sur des considérations "objet". Autrement dit, il sera bénéfique de détecter les objets dans la scène, et ainsi prévoir les zones d'occlusion ou de fort mouvement.

Extension du modèle gaussien généralisé au cas non spatialement stationnaire : on a vu que dans certains cas de figure, les performances restaient inchangées quels que soient les paramètres de la gaussienne généralisée modélisant le bruit de corrélation. Autrement dit, la distribution à modéliser n'est pas bien choisie, et mériterait d'être considérée comme non stationnaire spatialement. En effet, dans une image, la corrélation entre l'information adjacente et l'image originale n'est pas la même suivant les régions, et ils serait intéressant de considérer ce phénomène avec une distribution gaussienne généralisée ou avec un mélange de gaussiennes.

Application des métriques de qualité de l'information adjacente : L'étude des métriques mesurant la qualité de l'information adjacente proposée dans ce manuscrit s'en tient à des considérations théoriques. Il serait donc intéressant d'appliquer ces idées afin d'améliorer les performances débit-distorsion du schéma. Par exemple, on pourrait penser à développer une méthode de génération d'information adjacente dans laquelle l'erreur quadratique moyenne serait remplacée par une des métriques proposées.

Une optimisation du codeur ESSOR afin de tester les différentes méthodes proposées sur deux types de codeur : même si nous avons présenté des résultats débit-distorsion du schéma de codage vidéo distribué ESSOR, nous avons vu que ces performances n'étaient pas encore optimisées. Il faudrait pour cela se pencher sur chacun des modules de ce schéma et d'optimiser (quantificationdes trames WZ dans le domaine transformée, estimation du bruit de corrélation, etc.). Une fois le codeur optimisé, nous pourrions alors tester les différentes contributions de cette thèse sur le schéma ESSOR. Il serait intéressant d'observer le comportement des métriques d'estimation de la qualité de

l'information adjacente avec un décodeur LDPC, ou bien tester la modélisation du bruit de corrélation par des gaussiennes généralisées sur ce même décodeur LDPC dans le domaine des ondelettes.

Le codage vidéo distribué, quel avenir ?

Le codage vidéo distribué est un domaine de recherche pour le moins atypique. En effet, de part sa nouveauté, son potentiel et la beauté des résultats théoriques sous-jacents, il constitue un domaine de recherche très populaire et de nombreuses équipes de recherche travaillent à l'amélioration des performances de codage, ce qui fait que l'état de l'art, malgré la jeunesse du domaine, est déjà conséquent. Cependant, cette effervescence est en train de s'estomper de nos jours. On voit dans certaines reviews d'articles que certains chercheurs commencent à être sceptiques quant au potentiel du codage distribué. D'une part les résultats ne sont pas à la hauteur des attentes pour le moment, d'autre part l'argument de la diminution de la complexité à l'encodeur convainc de moins en mois. En effet, l'application phare initiale du codage vidéo distribué étant les systèmes à faible puissance de calcul (type téléphone portable), on peut facilement comprendre qu'avec les progrès d'efficacité des processeurs existants, les téléphones portables pourront soutenir très rapidement des calculs de plus en plus lourds.

Ce n'est pas pour autant qu'il faut se montrer défaitiste au sujet du codage vidéo distribué. En effet, si l'argument de la complexité ne pèse plus, il y aura toujours un avantage considérable que le codage vidéo distribué apportera : celui de supprimer tout besoin de communication entre les caméras à l'encodage. Il est fort probable qu'il faille attendre longtemps avant que les progrès technologiques viennent balayer cet argument. Une autre raison de se montrer optimiste quant à l'avenir du codage vidéo distribué est le formidable potentiel que celui-ci offre aujourd'hui. Pour chacun des modules, il est clair qu'il reste encore de fortes progressions à faire. Par exemple, les techniques de génération d'information adjacentes doivent être encore améliorées, et spécialement dans le sens des vues. Un gros enjeu du codage vidéo distribué est la modélisation de la corrélation qui doit savoir trouver les différentes stationnarités existantes. Enfin, si certains chercheurs pointent les limites du schéma de type Stanford, il reste néanmoins possible d'inventer d'autres schémas de codage, permettant de se rapprocher des conditions des théorèmes fondamentaux.

Contents

Introduction **39**

1 Distributed coding principles **45**
1.1 Distributed source coding . . . 46
1.1.1 Theoretical statement . . . 46
1.1.1.1 Definition and problem statement . . . 46
1.1.1.1.a Probability mass function and entropy . . . 46
1.1.1.1.b Rate and admissibility of the rate . . . 46
1.1.1.1.c Extension to the case of two correlated sources . . . 47
1.1.1.1.d Distortion . . . 48
1.1.1.2 Problem statement . . . 48
1.1.1.3 Lossless transmission . . . 48
1.1.1.4 Lossy transmission . . . 49
1.1.2 Applications . . . 51
1.2 Distributed video coding . . . 51
1.2.1 Prism Architecture . . . 52
1.2.1.1 Prism encoder . . . 53
1.2.1.2 Prism decoder . . . 54
1.2.1.3 Performance and related works . . . 54
1.2.2 Stanford approach . . . 54
1.2.2.1 Key frame coding . . . 55
1.2.2.2 WZ frame coding . . . 55
1.2.2.2.a Image classification . . . 55
1.2.2.2.b Transform . . . 56
1.2.2.2.c Quantization . . . 56
1.2.2.2.d Channel encoder . . . 56
1.2.2.2.e Side information generation . . . 57
1.2.2.2.f Channel decoder . . . 57
1.2.2.2.g Reconstruction . . . 57
1.2.2.2.h The drawbacks of the backward channel . . . 58
1.2.2.2.i Hash-based schemes . . . 58
1.2.3 Multiview distributed video coding . . . 58
1.2.3.1 Schemes . . . 58
1.2.3.2 Side information . . . 59
1.3 Conclusion . . . 59

I Rate distortion model and applications 61

2 Rate distortion model for the prediction error 63
2.1 Context . . . 64
2.2 Hypotheses and calculation . . . 65
2.3 Model validation . . . 67
2.3.1 Approximation for quantization distortion . . . 67
2.3.2 Decorrelation between the quantization and the motion/disparity estimation errors . . . 69
2.3.3 M_{d_1,d_2} does not depend on the quantization level . . . 70
2.3.4 Discussion about hypothesis validation . . . 71
2.4 Rate distortion model . . . 73
2.4.1 Results from information theory . . . 73
2.4.2 Proposed model . . . 75
2.5 Conclusion . . . 75

3 Applications of the rate-distortion model 77
3.1 Multiview schemes . . . 78
3.1.1 State-of-the-art . . . 78
3.1.2 Symmetric schemes . . . 80
3.1.3 Experimental validation . . . 84
3.2 Frame loss analysis . . . 88
3.2.1 Context . . . 88
3.2.2 Theoretical analysis . . . 89
3.2.3 Experimental validation . . . 91
3.3 Backward channel suppression . . . 92
3.3.1 Introduction . . . 92
3.3.1.1 Motivations and related problems of rate control at the encoder . . . 92
3.3.1.2 Existing rate estimation algorithms . . . 94
3.3.1.3 Hypotheses and main idea of the proposed approach . . . 95
3.3.2 Frame rate estimation . . . 95
3.3.2.1 Rate expression . . . 96
3.3.2.2 Homogeneous distortion inside the GOP . . . 96
3.3.2.3 Practical approach . . . 96
3.3.2.4 Experiments . . . 97
3.3.3 Bitplane rate estimation . . . 100
3.3.3.1 Wyner-Ziv frame encoding . . . 100
3.3.3.2 Proposed algorithm . . . 100
3.3.3.3 Experiments . . . 102
3.4 Conclusion . . . 104

II Side information construction 105

4 State-of-the-art of the side information generation 107
4.1 Estimation methods . . . 109
4.1.1 Interpolation . . . 109

4.1.2 Extrapolation . . . 114
4.1.3 Disparity . . . 117
4.1.4 Spatial estimation . . . 119
4.1.5 Refinement methods . . . 120
4.2 Fusion . . . 121
4.2.1 Problem statement . . . 121
4.2.2 Symmetric schemes . . . 121
4.2.3 Other schemes . . . 123
4.3 Hash-based schemes . . . 124
4.3.1 Definition of a hash-based scheme . . . 124
4.3.2 Hash information transmission . . . 124
4.3.2.1 Hash selection . . . 124
4.3.2.2 Hash compression . . . 125
4.3.3 Hash based side information generation methods . . . 126
4.3.3.1 Hash motion estimation / interpolation . . . 126
4.3.3.2 Genetic algorithm fusion . . . 126
4.4 Conclusion . . . 126

5 Essor project scheme 127
5.1 A wavelet based distributed video coding scheme . . . 128
5.1.1 Key Frame Encoding and Decoding . . . 128
5.1.2 Wyner Ziv Frame Encoding . . . 129
5.1.2.1 Discrete Wavelet Transform and quantization . . . 129
5.1.2.2 Accumulate LDPC coding . . . 130
5.1.3 Wyner-Ziv Frame Decoding . . . 132
5.1.3.1 Accumulate LDPC Decoding . . . 132
5.2 Proposed interpolation method . . . 134
5.2.0.2 Forward and Backward motion estimation . . . 134
5.2.0.3 Bidirectional Interpolation . . . 134
5.3 Experimental results . . . 136
5.3.1 Lossless Key frames . . . 136
5.3.2 Lossy Key frame encoding with H.264 Intra . . . 136
5.3.3 Lossy Key frame encoding with JPEG-2000 . . . 137
5.3.4 Interpolation error analysis . . . 138
5.3.5 Rate-distortion performances . . . 138
5.4 Conclusion . . . 139

6 Side information refinement 143
6.1 Generation of dense vector fields . . . 144
6.1.1 Motivations and general structure . . . 144
6.1.2 Cafforio-Rocca algorithm (CRA) . . . 145
6.1.2.1 Monodirectional refinement . . . 146
6.1.2.1.a Principle . . . 146
6.1.2.1.b First experiments . . . 147
6.1.2.2 Bidirectional refinement . . . 149
6.1.2.2.a Principle . . . 149
6.1.2.2.b First experiments . . . 151
6.1.3 Total variation based algorithm . . . 154

6.1.3.1 Monodirectional refinement . . . 155
6.1.3.1.a Principle . . . 155
6.1.3.1.b First experiments . . . 157
6.1.3.2 Bidirectional refinement . . . 157
6.1.3.2.a Principle . . . 157
6.1.3.2.b First experiments . . . 159
6.1.4 Experiments . . . 159
6.2 Proposed fusion methods . . . 164
6.2.1 Recall of the context . . . 164
6.2.2 Proposed techniques . . . 164
6.2.3 Experimental results . . . 166
6.3 Conclusion . . . 168

7 Hash-based side information generation 171
7.1 Proposed algorithm . . . 172
7.1.1 General structure . . . 172
7.1.2 Hash information generation . . . 173
7.1.3 Genetic algorithm . . . 175
7.2 Zoom on the three setting-dependent steps . . . 175
7.2.1 Initial side information generation . . . 175
7.2.2 Side information block distortion estimation . . . 177
7.2.3 Candidates of the Genetic Algorithm . . . 177
7.3 Experimental results . . . 179
7.3.1 First results . . . 179
7.3.2 Rate-distortion results . . . 180
7.4 Conclusion . . . 180

III Zoom on Wyner Ziv decoding 183

8 Correlation noise estimation at the Slepian-Wolf decoder 185
8.1 State-of-the-art: existing models . . . 186
8.1.1 Pixel domain . . . 186
8.1.1.1 Sequence level . . . 189
8.1.1.2 Frame Level . . . 189
8.1.1.3 Block level . . . 189
8.1.1.4 Pixel Level . . . 190
8.1.2 Transform domain . . . 190
8.1.2.1 Sequence level . . . 190
8.1.2.2 Frame Level . . . 192
8.1.2.3 Coefficient level . . . 192
8.1.3 Performance evaluation . . . 192
8.2 Proposed model: Generalized Gaussian model . . . 193
8.2.1 Definition and parameter estimation . . . 193
8.2.1.1 Moment estimation . . . 193
8.2.1.2 Maximum likelihood estimation . . . 194
8.2.1.3 Comparison . . . 194
8.2.2 Approach validation . . . 195

8.2.3 Experimental results . . . 197
8.2.3.1 Experimental setting . . . 197
8.2.3.2 Comparison in the offline setting . . . 197
8.2.3.3 Comparison in the online scenario . . . 198
8.2.3.4 Comparison between the offline and online settings . . . 198
8.2.3.5 Discussion . . . 199
8.3 A more complete study . . . 199
8.3.1 Motivations . . . 199
8.3.2 Experiments and results . . . 200
8.3.2.1 Experiments setting and results . . . 200
8.3.2.2 Discussion . . . 205
8.3.3 Conclusion . . . 205

9 Side information quality estimation 207
9.1 Motivations . . . 208
9.2 State-of-the-art . . . 208
9.2.1 PSNR metric . . . 208
9.2.2 SIQ . . . 209
9.3 Proposed metric . . . 210
9.3.1 Generalization of the SIQ . . . 210
9.3.2 A Hamming distance based metric . . . 211
9.4 Methodology of metric comparison . . . 212
9.5 Experimental results . . . 214
9.5.1 Common side information features . . . 214
9.5.2 The reasons why the PSNR is commonly used . . . 214
9.5.2.1 Experiment settings . . . 214
9.5.2.2 Discussion . . . 215
9.5.3 The limits of the PSNR . . . 217
9.5.3.1 Experiment settings . . . 217
9.5.3.2 Discussion . . . 217
9.6 Conclusion . . . 223

Conclusion 223

List of publications 234

Appendix - Compressed sensing of multiview images based on disparity estimation methods 237

Bibliography 262

Introduction

Since decades, video compression has been a main research topic which has mobilized many research groups and many industrials. From its initial goal which simply consists in reducing the rate necessary for the description of a video sequence, many other issues have risen depending on transmission, material or system power conditions. Indeed, whereas the purpose of all of these new paradigms remains to improve the compromise between a low rate and a high decoded quality, it is obvious that external conditions have a strong influence on adopted techniques or on more precise goals. For example, the video coding architecture would not be the same whether encoding and decoding is performed with a powerful system or not, or whether there is one camera or several.

So-called *classical* video compression (because more usual) aims at extracting inter frame correlation at the encoder. This approach thus relies on complex techniques (in terms of power requirements) such as motion estimation (or disparity estimation for multi-view sequences) in order to reduce the quantity of information to transmit to the decoder. This scheme is perfectly adapted to the following conditions: a compression performed on a powerfull station, and a light decoding with low-power systems (DVD player, TV broadcasting, etc.). However, whereas these configurations remain usually adopted, some new needs have risen in the last years. Indeed, more and more capture hardware systems need to perform video compression. Furthermore, more and more camera networks systems (such as videosurveillance) require non-complex compression algorithms and above all coding techniques which do not need communication between cameras (necessary in classical video coding since it is needed to extract the intercamera correlation).

Based on all these arguments, *distributed* video coding paradigm has appeared in early 2000's. This new paradigm proposes to shift all of the complex interframe comparisons to the decoder side. This idea is based on 30-year old theoretical results from Slepian and Wolf on one hand, and Wyner and Ziv on the other hand, which have stated that, under some specific conditions, two correlated sources could be encoded independently or jointly and transmitted with the same rate and the same distortion, as soon as the decoding is performed jointly.

These seductive theoretical results have led several research teams to develop distributed video coding schemes with the purpose (theoretically possible) to equal the performance of classical schemes such as MPEG-x, H.263, then H.264, etc. However, even if distributed video coding has been rapidly seen as a promising paradigm, the rate-distortion performance of current coders is far from the initial target. Indeed, several hypotheses of the founder theorems are not strictly verified and thus limit the efficiency of the existing codecs. Distributed video coding has nevertheless a lot of room for improvement since many modules can still be enhanced.

The European project DISCOVER has permit to several research teams to develop a complete distributed video coding scheme which is nowadays one of the most efficient and popular existing architectures. This scheme will be the starting point of most of the works presented in this thesis manuscript. That is why we draw, here, the main characteristics of this approach. The images of the sequence are divided into two types, the key frames and the Wyner-Ziv (WZ) frames, split as follows: one key frame, then one WZ frame, another key frame, and so forth. The key frames are independently encoded and decoded using intra codecs such as H.264 Intra or JPEG2000. These are also used at the decoder to generate a WZ frame estimation, called *side information*. The WZ frames are encoded independently with the classical source coding process: a transformation followed by a quantization. Then, instead of the entropy coder (usually adopted in classical source coding schemes) the output of the quantizer is processed with a channel encoder (LDPC or turbocodes), obtaining a systematic stream (a version of the input), and a parity stream (the redundancy information used to correct the channel errors). The idea consists in not transmitting the systematic information and in replacing it at the decoder by the side information generated with the key frames. Thereby, the parity information, initially designed to correct the channel error is transmitted in order to avoid the estimation errors. The WZ stream is then reconstructed and inverse transformed.

The original idea of using channel codes for compression is what makes distributed video coding original and attractive, but it is on the other hand what raises the largest number of limiting aspects and research works. Firstly, the system needs to know the correlation between the side information and the original WZ frame, yet, these two elements are not together available, neither at the encoder nor at the decoder. Moreover, the encoder needs to know the exact number of parity bits to send. That is why, the DISCOVER architecture (and almost all of the existing ones) make a progressive decoding by using a backward channel to request some more parity information as one goes along. It is one major limit of the system because it requires a hardly conceivable real-time transmission and decoding.

The second key element of this scheme is the side information generation task. Decoding performances strongly depend on the WZ estimation quality. That is why many works aims at enhancing the efficiency of motion/disparity estimation techniques.

The work conducted during this thesis led us to investigate many aspects of distributed video coding. First of all, we aimed at studying in detail the conditions of extending distributed video coding to multiview settings, which brings some new important questions, such as the disposition of the key and WZ frames in the time-view space, or the way of generating inter-view estimations and how to merge it with the temporal estimation so that the decoder has a unique side information. While proposing some solutions to these different problems, we have looked into several general aspects of distributed video coding (non specifically monoview or multiview), such as the improvement of temporal interpolation, a refinement of the correlation noise model, the backward channel suppression and a study of the side information quality metrics. Moreover, we have also studied other distributed video coding schemes by developping a hash-based scheme, and a wavelet-based coding architecture in collaboration with different research groups (LSS, IRISA and I3S).

Thereby, in this manuscript, we will present these contributions, their detailed context, purpose and results. These ones are organized in three parts, each of them corresponding to a different theme. The first part will present our contribution to improve the compre-

hension of the coder behavior in general, and its rate-distortion performance in particular. In a second part, we present some new improvements for the side information generation, and finally, in the third and last part, we make a zoom on the WZ decoder difficulties. In more details, the manuscript is organized as follows:

Chapter 1 - A distributed coding state-of-the-art: we will present the origins of distributed video coding through a rapid study of the distributed source coding techniques, and the two main approaches for distributed video coding. Moreover, we will detail the architecture of DISCOVER coder and its different still open problems. This chapter will not present a detailed state-of-the-art of each module because this will be done later in each chapter.

Part I - Proposal and applications of a rate-distortion model: In this part, the general behavior of a distributed video coding scheme is first analysed and modelled. Based on an original rate-distortion model, we will more precisely study the coder input (the frame classification), and the output with the error propagation phenomenon in case of frame losses. Finally, we propose an original solution to get rid of the backward channel. This first part contains two chapters:
Chapter 2 - A new rate-distortion model: we present here an original study which aims at modelling the WZ estimation error at the decoder. The obtained expression has a very simple and interesting structure which separates (mainly at high bitrates) the errors coming from the key frame quantization, and the errors coming from the motion estimation. This model is based on several hypotheses whose validity will also be tested in this chapter.
Chapter 3 - Applications of the rate-distortion model: in this chapter we describe three problems for which we have resorted to the proposed distortion model. The first of them corresponds to the image classification at the coder input. We detail all of the existing classifications in multiview configuration, and we then propose a new one involving a reduced number of reference frames, leading thus to a less complex encoding. Based on the proposed rate distortion model, we will determine the optimal decoding strategy (*i.e.*,WZ decoding order) of this scheme. Then, we will study the error propagation phenomenon in case of entire frame loss in monoview setting. We will observe the relative importance of the images depending on their position in the decoding order, and we will perceive some fundamental notions related to rate control at the encoder, such as the idea of not allocating an identical rate to all the WZ frames, and of taking into account their position in the sequence. Finally, we propose a new scheme allowing to get rid of the backward channel. Based on the proposed rate-distortion model, the rate control algorithm estimates the global frame rate and divided it between the bitplanes in function of the Hamming distance, contrary to the existing techniques which directly estimate the bitplane rates based on the entropy estimation.

Part II - Side information generation: in this part, we will exclusively study the WZ estimation process at the decoder, motivated by the observation that distributed video coding performance strongly depends on side information quality. After a detailed review of the existing techniques in the literature, we present the interpolation algorithm designed in collaboration with other research teams of the ESSOR project (see Chapter 5 for more details). Then, we detail the proposed dense (one vector per pixel) interpolation

algorithms as well as the proposed methods for the fusion of the inter-view and temporal estimations. Finally, we will present an original hash-based scheme.
Chapter 4 - State-of-the-art : we present here all the problems related to the side information such as the estimation methods (interpolations, extrapolations, etc.), then the fusion of several estimations (for multiview setting), and finally the existing hash-based schemes which helps the decoding process by sending well-chosen WZ informations.
Chapter 5 - ESSOR interpolation: this chapter details the interpolation technique proposed within the ESSOR project. We also detail the developed coder in which this algorithm has been integrated, and we show some rate-distortion results.
Chapter 6 - Side information refinement : the techniques detailed in this chapter are based on the idea that the savings of the interpolation vector number was not justified isince the WZ estimation is performed at the decoder, and on the fact that it was possible to describe the motion/disparity with a dense field (one vector per pixel). We have proposed a family of vector field refinement methods, starting from the DISCOVER interpolation structure, and adding two refinement steps, each of them performed by two possible adapted techniques: the modified Cafforio-Rocca algorithm [Cafforio, Rocca, 1983] and the Miled one [Miled *et al.*, 2009] (based on the total variation). Finally, in this chapter, we propose three original fusion methods, performing a linear combination between the pixels instead of a binary choice as usually done in the literature.
Chapter 7 - Hash-based scheme: aware of the fact that the decoder does not have all of the informations necessary for a perfect WZ frame estimation, some solutions have been proposed to send some so-called *hash* information, which corresponds to a localized and well-chosen WZ frame description, in order to enhance the side information generation process at the decoder side. In this chapter we propose a novel approach for generating and selecting the hash information, and moreover we extend the algorithm developed by Yaacoub *et al.* [Yaacoub *et al.*, 2009a] to a multiview configuration.

Part III - Zoom on the Wyner-Ziv decoder: In this part, we study two problems related to the turbo decoding process. Firstly we propose to refine the correlation noise modelling, and then we focus on the metrics used to estimate the side information quality. Thiw part contains two chapters:
Chapter 8 - Correlation noise estimation: in this chapter we will present a detailed review of the existing techniques which aim at modelling the correlation noise. The conclusion of this review is that the finer the model is (and the closer to the true distribution the estimated probability density function is), the better the performance. As a consequence, we have proposed to use a Generalized Gaussian model instead of the commonly adopted Laplacian one. The obtained rate-distortion results are mitigated. Whereas the proposed refinement mostly leads to a decoding efficiency enhancement, their exist some cases for which the performance remains unchanged. To better understand this behaviour, we propose a more advanced study which will be detailed at the end of this chapter.
Chapter 9 - Side information quality estimation: when a side information generation method is tested, it is commonly evaluated with the PSNR. Yet, Kubasov [Kubasov, 2008] has shown that this metric could lead, in some situations, to a wrong estimation of its quality. In this chapter, we propose to extend his study and try to understand when the PSNR is suitable, and when this measure may present some reliability limits. Furthermore, we propose new metrics and test for each of the studied situations the reliability of the proposed metrics, more adapted to the turbodecoder behavior.

Appendix - Compressed sensing of multiview images based on disparity estimation methods: beside my PhD, I have been led to work on other topics, not integrated in this manuscript, because they are quite far from distributed video coding paradigm. They deal with a very famous subject: the compressed sensing, and we propose to extend some existing methods in video to multiview images and sequences by applying some of the disparity estimation methods described in this manuscript. The common point with distributed video coding, that inspired our contributions, is the necessity to take into account *at the reconstruction* the correlation that exists between frames, either in multi-component images, or multiview sequences. As for distributed video coding, the estimation of the motion and/or disparity fields is based on reconstructed frames and does not need to take into account the rate of the resulting vector field. This enables the use of dense estimation methods, and is one of our original contributions, together with different algorithms for reconstructing images and displacement fields iteratively. This appendix contains all of the published articles related to compressed sensing.

In order to implement and evaluate all of the contributions described here, we have developed on one hand a multiview extension of the Discover coder, and on the other hand a complete wavelet-coder within the Essor project.
Moreover, we precise here, that this thesis was done as part of two projects: Essor, french ANR project (constituted by the LSS, the IRISA, I3S and Telecom ParisTech) and Cedre, a franco-lebanese project in collaboration with the Holy-Spirit University of Kaslik (USEK).

Chapter 1

Distributed coding principles

In this chapter, we first introduce the origin of distributed video coding. After recalling some basic notions in information theory, we present the fundamental results of Slepian and Wolf in case of lossless coding, and of Wyner and Ziv in case of lossy coding. Then we explain how this theory of distributed source coding has been brought into practice 30 years after its publications.

*The approach of distributed source coding in video compression has attracted much interest and we present in Section 1.2 the two main existing architectures (*Prism *and Stanford). Since all the contributions exposed in this manuscript thesis have been proposed in the framework of a distributed video coding scheme inspired by the Stanford approach, we explain in detail how it operates, and for every module, we list the open questions and we briefly introduce how we have tried to answer them in the next chapters.*

Contents

1.1 Distributed source coding **46**
1.1.1 Theoretical statement 46
1.1.2 Applications 51
1.2 Distributed video coding **51**
1.2.1 Prism Architecture 52
1.2.2 Stanford approach 54
1.2.3 Multiview distributed video coding 58
1.3 Conclusion **59**

1.1 Distributed source coding

In this section we briefly introduce the principles underlying the distributed source coding (DSC) paradigm, which comes from two fundamental results of information theory stated in the 1970's and put in application to video transmission only recently. In Section 1.1.1, we first present the theoretical background of DSC, and then, in Section 1.1.2 we present the main practical applications.

1.1.1 Theoretical statement

Slepian-Wolf and Wyner-Ziv works aim at studying the classical problem of encoding and decoding two correlated sources X and Y (the transmission is performed over a lossless channel). Before presenting these two surprising and important theorems in Paragraphs 1.1.1.3 and 1.1.1.4, we introduce some useful notions taken from information theory, for a better understanding of the following.

1.1.1.1 Definition and problem statement

1.1.1.1.a Probability mass function and entropy

Let $\mathscr{A} = \{K_1, K_2, ..., K_A\}$ be a set of A elements, and let X be a discrete random variable taking its values in $\mathscr{A}$. The *probability mass function (pmf)* of X is defined by

$$p_X(x) = \text{Prob}\left[X = x\right], \quad x \in \mathscr{A}. \tag{1.1}$$

If $\mathbf{X} = (X_1, X_2, ..., X_n)$ is a vector of n independent realizations of X, the pdf definition of $\mathbf{X}$ becomes:

$$p_{\mathbf{X}}(\mathbf{x}) = \text{Prob}\left[\mathbf{X} = \mathbf{x}\right], \quad \mathbf{x} = (x_1, x_2, ..., x_n) \in \mathscr{A}^n. \tag{1.2}$$

Based on the intuitive idea that a rare element brings more information than a more probable one, Shanon has proposed a definition of the *self-information* of a symbol $x \in \mathscr{A}$:

$$I(x) = -\log_2\left(p_X(x)\right).$$

The *entropy* (in bits) of a discrete random variable X is a measure of the amount of uncertainty one has about the values of the variable. It is defined as the self-information average of the elements in the set $\mathscr{A}$:

$$H(X) = -\sum_{i=1}^{A} p_X(K_i)\log_2\left(p_X(K_i)\right). \tag{1.3}$$

An important property of entropy is that it is maximized when all the messages in the message space are equiprobable.

1.1.1.1.b Rate and admissibility of the rate

The entropy is not a simple measure of uncertainty, it is also one theoretical bound for the rates as stated in an important theorem. Before writting it, let us recall some notions of source coding. Firstly, an *encoder*, $\mathscr{C}(n, M)$ associates the input vector $\mathbf{x} = (x_1, x_2, ..., x_n)$ to an integer of the set $\mathscr{M} = \{1, 2, \ldots, M\}$. Then, after the channel, the *decoder*, $\mathscr{D}(n, M)$,

associates this integer of $\mathscr{M}$ to a vector $\hat{\mathbf{x}} \in \mathscr{A}^n$. In this configuration, this couple encoder-decoder is related to a *rate*, $R = \frac{1}{n}\log_2(M)$ which defines the information unit per element sent by the transmitting source. Based on these definitions, a rate R is said *admissible* if for all $\varepsilon > 0$, there exists a n, an encoder $\mathscr{C}\left(n, [e^{nR}]\right)$ and a decoder $\mathscr{D}\left(n, [e^{nR}]\right)$ such as $\text{Prob}\left[\hat{\mathbf{X}} \neq \mathbf{X}\right] < \varepsilon$.
Based on this notion, one can state the following theorem.

Théorème 1 *If $R > H(X)$, R is admissible.*

In other words, the entropy constitutes the lower bound of the set of admissible rates.

1.1.1.1.c Extension to the case of two correlated sources

The extension of the previous notions to two correlated sources leads to similar definitions. Indeed, if X and Y are two correlated random variables taking their values respectively in $\mathscr{A}_X = \{K_1, K_2, ..., K_{A_X}\}$ and $\mathscr{A}_Y = \{K_1^{'}, K_2^{'}, ..., K_{A_Y}^{'}\}$, their joint pmf is defined by:

$$p_{XY}(x, y) = \text{Prob}\left[X = x, Y = y\right], \quad x \in \mathscr{A}_X, \quad y \in \mathscr{A}_Y. \tag{1.4}$$

In the same manner, if $\mathbf{X} = (X_1, X_2, ..., X_n) \in \mathscr{A}_X^n$ and $\mathbf{Y} = (Y_1, Y_2, ..., Y_n) \in \mathscr{A}_Y^n$, their joint pmf is:

$$p_{\mathbf{XY}}(\mathbf{xy}) = \text{Prob}\left[\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}\right] = \prod_{i=1}^{n} p_{x_i}(y_i), \tag{1.5}$$

with $\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathscr{A}_X^n$, $\mathbf{y} = (y_1, y_2, ..., y_n) \in \mathscr{A}_Y^n$. The transmission of X and Y is performed by using two encoders and two decoders. Similarly to the case of a unique source, the couple of rates (R_X, R_Y) is said admissible when there exist encoders and decoders which enable a perfect recontruction of both sources.

Based on the joint pmf, one can define the *marginal distributions*:

$$p_X(x) = \sum_y p_{XY}(x, y) \tag{1.6}$$

$$p_Y(y) = \sum_x p_{XY}(x, y) \tag{1.7}$$

and the *conditional distributions*, drawing the probability of a source when the other is known:

$$p_{X|Y}(x) = \frac{p_{XY}(x, y)}{p_Y(y)} \tag{1.8}$$

$$p_{Y|X}(y) = \frac{p_{XY}(x, y)}{p_X(x)}. \tag{1.9}$$

Consequently, the joint entropy can be defined by

$$H(X, Y) = -\sum_x \sum_y p_{XY}(x, y) \log_2\left(p_{XY}(x, y)\right) \tag{1.10}$$

and the conditional entropy by

$$H(X|Y) = -\sum_y p_Y(y) \sum_x p_{X|Y}(x|y) \log_2\left(p_{X|Y}(x|y)\right) \tag{1.11}$$

and identically for $H(Y|X)$.

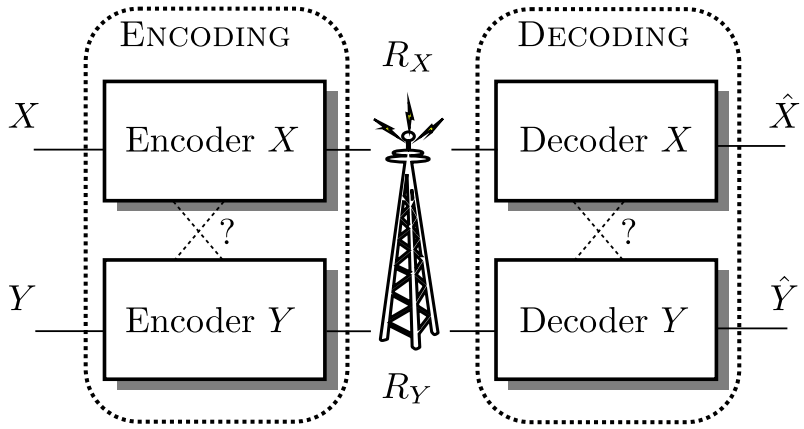


Figure 1.1: Two correlated source transmission scheme. R_X and R_Y are respectivly the rates for the two sources X and Y. The dashed lines between the encoders and between the decoders correspond to potential communication links between them.

1.1.1.1.d Distortion

When the reconstruction is not perfect, a fidelity criterion called *distortion* is commonly introduced in order to measure the difference between $\hat{\mathbf{X}}$ and $\mathbf{X}$:

$$d = \frac{1}{n}\sum_{k=1}^{n} D(X_k, \hat{X}_k) \tag{1.12}$$

where $D(x, \hat{x})$ is a given distortion function defined on $\mathscr{A} \times \hat{\mathscr{A}}$. This distortion is used to define the *rate-distortion (RD) function* which gives for a given distortion d, the minimum rate R allowing a transmission with a reconstruction at this distortion.

1.1.1.2 Problem statement

We present here the hypotheses of two fundamental theorems presented in the following, which have initiated the DSC paradigm. The problem, summarized in Figure 1.1, deals with the conditions of coding two correlated sources X and Y. Firstly, they are encoded with their own separate encoder. Then they are transmitted over a lossless channel, with a respective rate of R_X and R_Y. Then, they are decoded and we denote by $\hat{X}$ and $\hat{Y}$ their reconstructed version. In case of lossy compression, $\hat{X}$ and $\hat{Y}$ do not entirely recover X and Y, while for lossless transmission we have $\hat{X} = X$ and $\hat{Y} = Y$.
The purpose of the theoretical study presented in the following is to determine the rate-distortion optimal conditions for this transmission in several configurations. These configurations differ on whether the knowledge of the other source is available or not at the encoder and/or at the decoder (dashed-line in Figure 1.1).

1.1.1.3 Lossless transmission

In 1973, Slepian and Wolf (SW) [Slepian, Wolf, 1973], studied the previously introduced problem (Section 1.1.1.2), in case of lossless transmission, *i.e.*,$\hat{X} = X$ and $\hat{Y} = Y$. They have given the *admissible rate region*, *i.e.*,the set

$$\{(R_X, R_Y) \text{ such as } R_X \text{ and } R_Y \text{ are admissible}\}$$

for several different configurations, which depend on whether the encoders and the decoders have access or not to the information about the other source. A first result, stated by Theorem 1, is that the admissible rate regions for a lossless transmission at least contains the set $\{(R_X, R_Y), \quad R_X \geq H(X) \text{ and } R_Y \geq H(Y)\}$. When the encoders or the decoders have the knowledge of the other sources, this minimum admissible rate region is extended. It is not the point here to detail all of Slepian-Wolf study, thus we only give two particular and interesting results:

- The **classical coding** is when the two encoders and the two decoders are able to use the information of the other source. For this configuration, if the source X is transmitted with a rate R_X, the source Y can be transmitted with a rate $H(X,Y) - R_X$, without having any loss at the decoder. In other words, the admissible rate region is

  $$R_X + R_Y \geq H(X,Y).$$

  This result can be observed in Figure 1.2 (a).

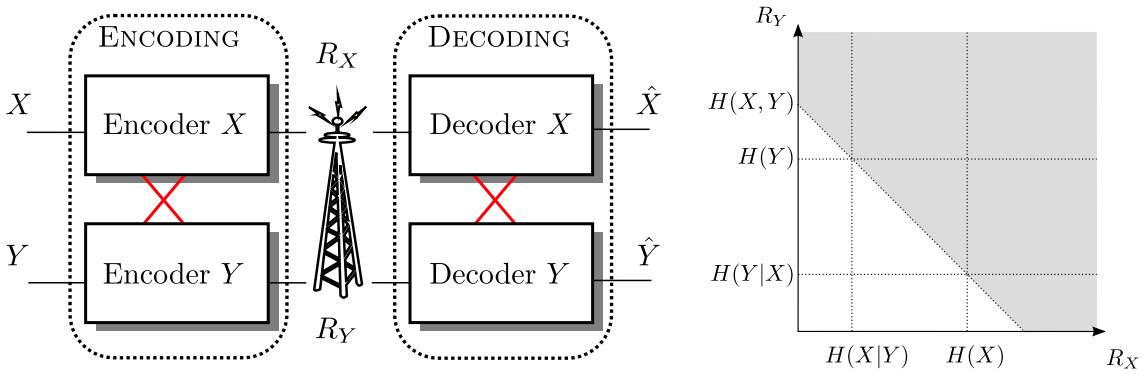
- The **distributed coding** is when the encoding of X and Y is, this time, performed *independently*, while the decoding is still done *jointly*. For this case, Slepian and Wolf stated that the admissible rate region has surprisingly the same lower bound. In other words, if the source X is transmitted with a rate R_X, the source Y can still be transmitted with a rate $H(X,Y) - R_X$ without any loss. One can also observe this important results on Figure 1.2 (b) which presents on the right the corresponding admissible rate region.

Therefore, Slepian and Wolf have stated in their paper that it is one and the same thing from the point of view of rate performance, to encode two correlated sources jointly or independently (for lossless transmission), while the decoders have the knowledge of both sources and of the correlation model.

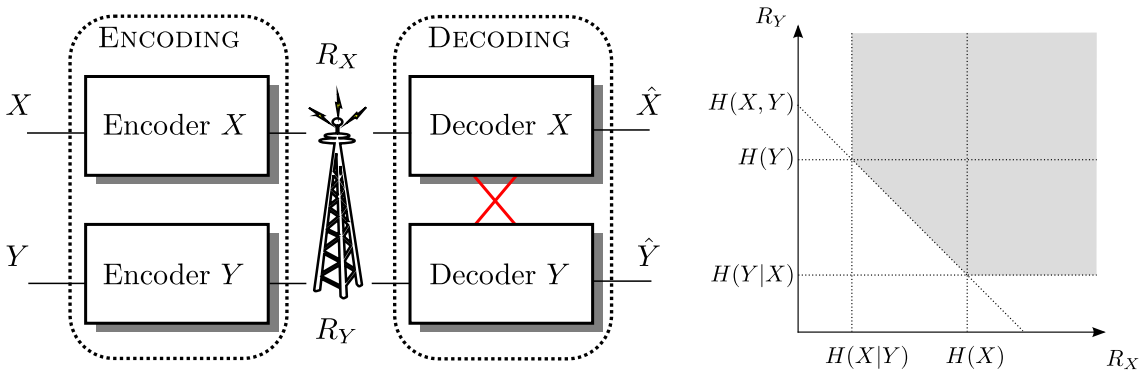
This theorem was the starting point of many papers. First works have rapidely risen in the 1970's, with Wyner who has used the Slepian and Wolf theorem in order to investigate multiple-user communication [Wyner, 1974], and extension to three sources independently encoded [Wyner, 1975]. In 1975, Cover has proven the Slepian and Wolf theorem in case of ergodic sources [Cover, 1975]. Even recently, distributed source coding for lossless transmission has been investigated, for example with an application to satellite communications [Yeung, Zhang, 1999], or for more involved lossless source coding networks (more than two sources, zig-zag networks, etc.) [Stankovic *et al.*, 2006].

1.1.1.4 Lossy transmission

In 1976, Wyner and Ziv (WZ) have extended the Slepian and Wolf theorem to lossy transmission [Wyner, Ziv, 1976], *i.e.*,when some information loss is allowed in the communication process. Instead of the admissible rate region Wyner and Ziv studied the rate distortion function for the same configuration. They have proven that if the distortion measure is the mean square error (MSE), and if the two sources are jointly Gaussian, the rate distortion function is identical for joint and independent encoding since the decoding is performed jointly. In other words, under some conditions on the pdf of the sources, distributed source coding can achieve the same performance as classical coding in case of lossy transmission.



(a) Classical coding scheme and the corresponding admissible rate region (gray area).



(b) Distributed coding scheme and the corresponding admissible rate region (gray area).

Figure 1.2: Results of Slepian-Wolf study for classical and distributed coding. The red links indicate the communications allowed during the encoding/decoding.

Several important works, based on Wyner and Ziv results, have been conducted soon afterwards. Berger, in 1977 [Berger, Longo, 1977] introduced the lossy version of non-asymmetric Slepian-Wolf scheme, *multiterminal (MT) source coding*. Research for lossy DSC is still nowadays very active. Indeed, a lot of important problems are still open, as the lossy MT source coding problem with two non-jointly Gaussian sources, as it was investigated in [Bassi *et al.*, 2009] for Bernouilli-Gaussian correlation.

1.1.2 Applications

The Slepian and Wolf paper [Slepian, Wolf, 1973] does not present how to reach the proven rate bounds. First practical solutions were brought by Wyner [Wyner, 1974] who proposed the use of linear channel codes. This was the beginning of many solutions which adopt a "syndrom-based" approach by using channel coder for data compression. The two channel codes which are mainly used are the low-parity-density-check codes (LDPC) [Liveris, 2002] [Varodayan *et al.*, 2005] and the turbocodes [Garcia-Frias, Zhao, 2001] [Aaron, Girod, 2002]. The turbocodes were proposed by Berrou *et al.*. [Berrou *et al.*, 1993] [Berrou, Glavieux, 1996]; the reader can refer to a clear tutorial on turbocodes in [Ryan, 1997] for more precisions.

Practical WZ schemes can be realized by using a quantization and a SW coder. The first applications, in 1999, proposed to combine these two processes. The resulting solution is called, Distributed source coding using syndromes (DISCUS) and is detailed in [Pradhan, Ramchandran, 1999] [Pradhan, Ramchandran, 2003]. The coding scheme proposed by Pradhan and Ramchandran is an asymmetric scheme, *i.e.*,one source Y is encoded alone (at a rate of $H(Y)$) and is used as *side information* only at the decoder to help the other source decoding X, and then to allow a transmission of X at a rate of $H(X|Y)$ theoretically. A more efficient solution is to make a quantization for the rate-distortion control followed by a SW coder, which plays the role of the entropy coder. The SW coder uses linear channel codes. A solution proposed by Yang *et al.*. in [Yang *et al.*, 2008] allows to come very close to the bounds in case of two jointly Gaussian sources coding. This method is based on a trellis-coded quantization and an efficient channel coding (with LDPC or turbocodes).

1.2 Distributed video coding

Both works of Slepian-Wolf and Wyner-Ziv have stated that it was possible, under certain conditions, to avoid the inter-source correlation extraction at the encoder without any loss in performance. If we consider the different frames of a video sequence as belonging alternatively to two correlated sources, one can immediately use these theoretical results for removing the very complex motion estimation between the frames at the encoder, without reducing the rate-distortion performance. On the contrary, in the case of distributed video coding, the comparison between frames is performed at the decoder. A reduction of the encoding complexity could be interesting for any kind of low-power systems, as videosurveillance, cellphone, etc. Moreover, in multicamera systems, a distributed coding approach could permit to avoid all the communications between cameras, needed by classical interframe coders [Guillemot *et al.*, 2007].

Then, first practicle implementations of *distributed video coding (DVC)* or WZ video

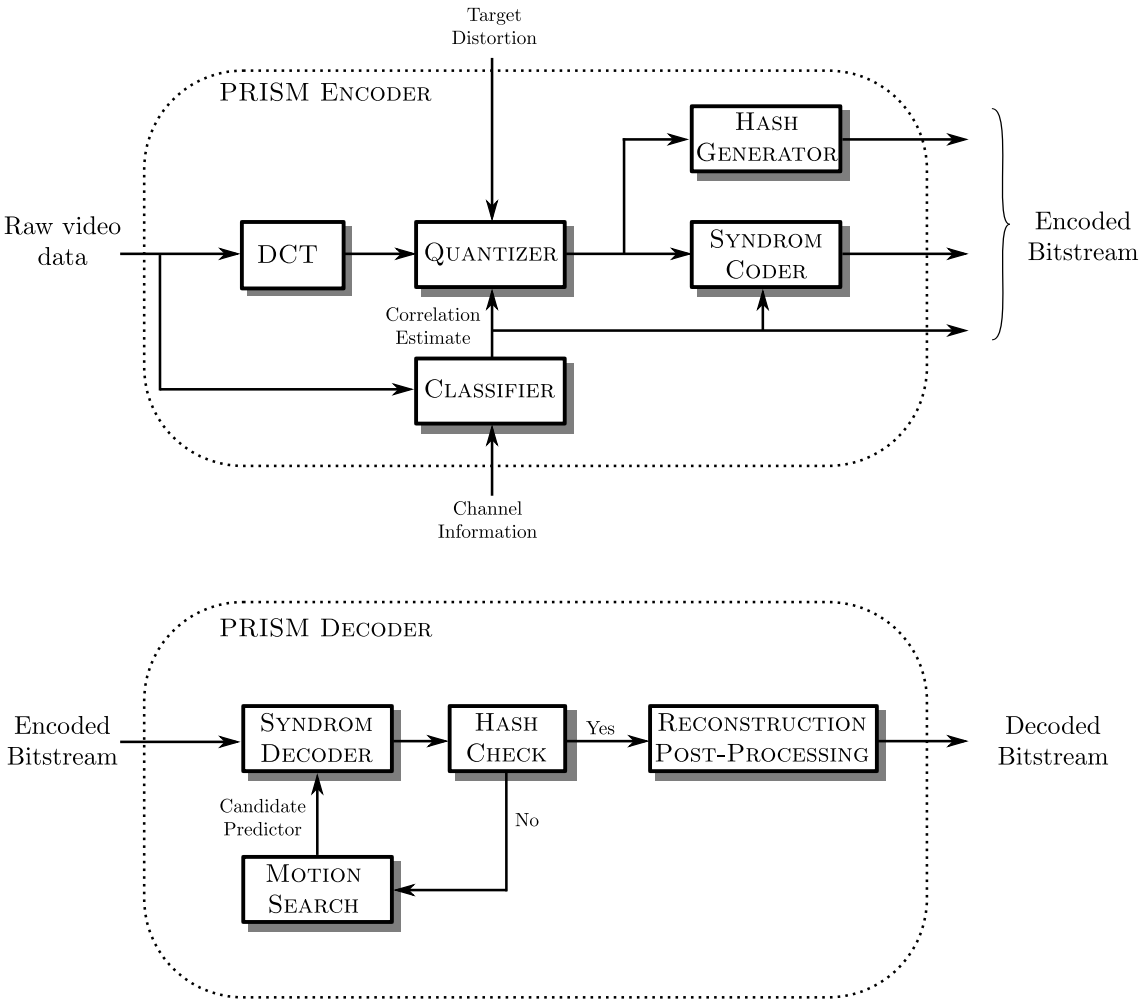


Figure 1.3: Prism architecture.

coding, appeared 30 years after the theory but soon after first WZ source coding schemes, in 2002 with two different approaches: the Prism architecture [Puri, Ramchandran, 2002] (detailed in Section 1.2.1) and the Stanford scheme [Aaron *et al.*, 2002] (detailed in Section 1.2.2). All the works proposed in this manuscript thesis are based on the Stanford approach, that is why, in the following, we give more importance to the techniques involved by this DVC scheme.

1.2.1 Prism Architecture

The Prism ("Power-efficient, Robust, hIgh compression, Syndrom-based Multimedia coding") architecture [Puri, Ramchandran, 2002][Puri, Ramchandran, 2003] was proposed in 2002. An evolved version of the coder has been implemented in 2007 (described in [Puri *et al.*, 2007]), and it is the one we have chosen to present here since it is better performing. The general scheme is summarized in Figure 1.3.

1.2.1.1 PRISM encoder

The frame is divided into blocks of size 8×8. Each block is processed with the coding scheme summarized in Figure 1.3, whose different steps are detailed in the following.

- **Transform:** the block is firstly transformed using a discrete cosinus transform (DCT). The outpout of this block is a one-dimensional vector, which contains the 64 coefficients arranged after a zig-zag scan on the two dimentionnal transformed 8×8 block.

- **Quantization:** the coefficients are then quantized using a scalar quantization mainly inspired by H.263+ one [Cote *et al.*, 1998].

- **Classification:** at the same time, the encoder performs a classification on the blocks, and more precisely on its bitplanes. This step is the most important one in the PRISM architecture because it is where the WZ approach rises. The purpose of this step is to determine which bitplane to transmit, and whether to WZ encode or to entropy encode. Moreover, this step aims at choosing the class for the block which corresponds to the level of correlation with the side information. The SI is called the reference block, and is obtained from different ways depending on the computation capacity of the encoder. If the encoder is powerful, a motion estimation is performed in order to find the most similar block in the previous frame (in that case, the obtained motion vector is transmitted). For low-power encoders, the reference block is simply the block which has the same location in the previous reference image. Having this reference block, the encoder compares the number of similar most significant bits in the bitplane decomposition of the current and reference blocks for each coefficient. The most significant bits which are identical in the block and its side information decomposition are not transmitted (because they will be recovered at the decoder). On the other hand, the remaing bits are either WZ encoded with a channel encoder (for the most significant of them) or simply entropy coded (for the least significant ones). Moreover, based on the sum of squared differences (SSD)[1] between the reference and the current block, the encoder determines an index i which indicates the class of Laplacian correlation noise which would help for the decoding.

- **Syndrom encoding:** as explained in the previous item, the transmitted bitstream is either WZ or entropy encoded. The adopted entropy coder is similar to the one adopted in some video compression standards [Cote *et al.*, 1998]. The channel coder is not an LDPC or turbocode because of the small length of the bitstream. The adopted channel encoder multiplies the input bitstream by a parity matrix (which depends on the correlation noise class i) and use the BCH [Macwilliams, Sloane, 1977] block codes, efficient for small-length bitstreams.

- **Hash generation:** in order to help the prediction at the decoder, a hash information is generated at the encoder. In the PRISM scheme, the hash is a CRC (Cyclic Redundancy Check) checksum of size 16 bits. This represents a "signature" of the original block which is used at the decoder to test the reliability of the prediction.

[1] The SSD between two vectors $\mathbf{x} = (x_i)_{i=1\ldots n}$ and $\mathbf{y} = (y_i)_{i=1\ldots n}$ is $\sum_{i=1}^{n}(x_i - y_i)^2$

1.2.1.2 Prism decoder

The decoder scheme is shown in Figure 1.3. In the following we describe each module used for the decoding of the 8×8 blocks.

- **Side information (SI) generation:** the purpose of this step is to find the best prediction of the block. The decoder performs a motion search, in order to generate a set of candidates. Each possible prediction is then decoded, *via* the rest of the decoding chain. The selected side information is the one whose decoded version satisfies the hash check module.

- **Syndrom decoding:** the syndrom decoding consists in two steps. Firstly, the bits which were entropy and WZ coded are decoded. Secondly, the decoder finds the closest codeword to the side information within the specified coset. This step is quite complex, and a less complex suboptimal algorithm [Fossorier, Lin, 1995] has been proposed with a loss of $0.2 - 0.3$ dB.

- **Hash check:** at this step, the checksum of the previously decoded block is calculated and compared to the transmited hash information. If it does not correspond, the decoding restarts with another candidate (given by the initial motion search).

- **Reconstruction, post-processing** once the quantized codeword recovered, a predictor is used to estimate the best reconstructed block in the sense of the mean square error (MSE). The reconstruction is then inverse transformed in order to obtain the decoded block.

1.2.1.3 Performance and related works

The experiments shown in [Puri *et al.*, 2007] state that the Prism architecure allows to approach the H.263+ inter frame coder performance for some test sequences. These performances were theoretically analyzed in [Majumdar *et al.*, 2005], and it was confirm that Prism architecture could perform a good compression for sequences containing slow and easily estimated motion, but less acceptable efficiency for more complex sequences, as *football*. An open-source implementation of this architecture was proposed by Fowler in 2005 [Fowler, 2005].
The main drawback of this coding scheme is that the proposed approach is not stricly distributed since the encoder needs a reference block, and then performs an inter frame comparison.

1.2.2 Stanford approach

At the same time, in 2002, a research group at the Stanford university proposed another approach for practical WZ video coding [Aaron *et al.*, 2002]. They have chosen to adopt a frame approach (contrary to the block-based Prism architecture) by splitting the sequence into two types of frames (which alternate along the time): the *key (K) frames* and the *Wyner-Ziv (WZ) frames*, and encoding these frames independently. The K frames form a Y source which is encoded/decoded alone, and the WZ frames constitute a X source which is encoded alone and decoded thanks to the side information given by Y.
One of the most popular Stanford scheme extensions was proposed by the European project

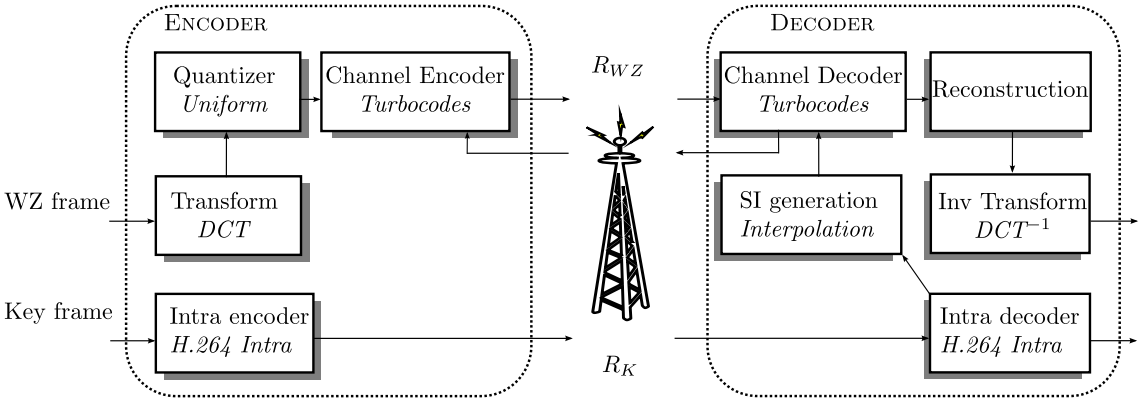


Figure 1.4: Generic Stanford architecture. In italic, the corresponding Discover approach.

Discover [DISCOVER-website, 2005]. In the following we detail the encoding and decoding process of the general Stanford scheme (summarized in Figure 1.4) and at the same time we specify the Discover approach (The Discover blocks are given in italic in Figure 1.4) and we detail block by block the various techniques proposed in the literature. Note that for some specific topics, a detailed state-of-the-art is proposed later in the manuscript, more precisely when we explain our contributions related to these topics.

1.2.2.1 Key frame coding

The key frame coding is relatively simple since it is performed with an intra frame coder. Some solutions involve a DCT-based intra codec such as the H.263+ codec [Aaron *et al.*, 2002] [Girod *et al.*, 2005], or H.264 Intra [Brites *et al.*, 2006b]. Some other works use a wavelet-based approach and use the JPEG-2000 still image codec for key frame compression, as explained in [Guillemot *et al.*, 2007].

1.2.2.2 WZ frame coding

1.2.2.2.a Image classification

The sequence is divided into two types of frames: the key frames and the WZ frames. A set of one K frame followed by n WZ frames is called a *Group of Pictures (GOP)*. The size of the GOP, $n+1$, is fixed in the majority of the works. If the GOP size is small (2), the estimation of the WZ frame would be of a better quality, but if the GOP size is larger (4, 16, etc.) the number of K frames decrease and the complexity too (because a K frame is more complex to encode than a WZ frame).
The image classification issue brings two fundamental questions. The first one concerns the determination of the optimal GOP size. In [Ascenso *et al.*, 2006], Ascenso *et al.* proposed a solution at the encoder for adapting the GOP size to the motion activity in the sequence, *i.e.*,a high motion activity would make the GOP size decrease, while the absence of high motion would lead to a larger GOP size.
Secondly, a large GOP size (greater or equal to 4) leads us to wonder about the optimal WZ frame decoding order. An empirical solution has been proposed in [Aaron *et al.*, 2003]. In Section 3.1, we propose a theoretical study which aims at determining the best decoding

order in case of a GOP length of 4.

1.2.2.2.b Transform

First solutions in DVC did not involve any transform and thus directly process the WZ frame in the pixel domain [Aaron *et al.*, 2002] [Girod *et al.*, 2005] [Ascenso *et al.*, 2005a] [Brites *et al.*, 2006a] [Morbee *et al.*, 2007].

Later, the idea of working in the transform-domain has appeared to be interesting since it allows to improve the performance without adding any sensible complexity. Almost all of the proposed solutions adopt the 4×4 integer DCT [Aaron *et al.*, 2004b] [Brites *et al.*, 2008]. The output of this module is then a matrix whose rows correspond to the 16 coefficients taken in the zig-zag order [Wiegand *et al.*, 2003], and whose columns correspond to the coefficients (image size divided by 16) taken in the raster order. These are also the solutions adopted by the Discover scheme as it can be seen in Figure 1.4.

Some other approaches prefer a wavelet transform, such as Guo *et al.* [Guo *et al.*, 2006a] [Guo *et al.*, 2006b] but they are far less numerous than the DCT based schemes.

1.2.2.2.c Quantization

In Discover, the quantization of the coefficients is done with a classical linear quantization (with a dead zone for AC coefficients) on 2^{m_b} levels, where m_b is the number of bits used for the description of the band b. The number of levels depends on the frequency index of the band, in order to describe the most significant bands (the first ones in the zig-zag order) more accurately. In Discover, this number of levels is given by 8 predetermined quantization points [Brites *et al.*, 2006b] (inspired from [Aaron *et al.*, 2004b]). This matrix, given in Table 1.1 presents the number of levels (2^{m_b}) depending on the band for 8 quantization points (called *quantization index* QI); QI=1 corresponds to low bitrate while QI=8 corresponds to high bitrate.
Having the number of levels, the encoder calculates the quantization step using the maximum band value (for the frame). This maximum value is transmitted to the decoder.

Table 1.1: WZ matrix setting 8 quantization points. For each QI, it is given the number of levels for the 16 bands.

band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
QI 1	16	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
QI 2	32	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
QI 3	32	8	8	4	4	4	0	0	0	0	0	0	0	0	0	0
QI 4	32	16	16	8	8	8	4	4	4	4	0	0	0	0	0	0
QI 5	32	16	16	8	8	8	4	4	4	4	4	4	4	0	0	0
QI 6	64	16	16	8	8	8	8	8	8	8	4	4	4	4	4	0
QI 7	64	32	32	16	16	16	8	8	8	8	4	4	4	4	4	0
QI 8	128	64	64	32	32	32	16	16	16	16	8	8	8	4	4	0

1.2.2.2.d Channel encoder

After the quantization, the WZ frames bitstream is channel encoded, obtaining two types of data: the systematic and the parity information. Originally, a channel encoder produces parity information in order to be able to correct at the decoder the errors in the systematic information. In distributed source coding based on channel codes, the systematic information is not transmitted, but replaced at the decoder by a side information (see next sections). Only a part of the parity information is transmitted to the decoder in order to

correct the side information error.

The existing solutions use either LDPC codes [Xu, Xiong, 2006] or, in the majority of the cases, the turbocodes [Aaron *et al.*, 2002] [Dalai *et al.*, 2006]. Guillemot *et al.*, in [Guillemot *et al.*, 2007] give a comparison between turbocodes and LDPC performance for WZ video coding, and they show that LDPC based schemes slightly outperform turbocodes based ones (with a very small gap).

1.2.2.2.e Side information generation

At the decoder side, the WZ frame is firstly estimated by the side information generation module. This estimation is performed using several kinds of techniques: interpolation, extrapolation, etc. We propose in Chapter 4 a detailed overview of the techniques existing in the literature.

It is important to note that DVC coding performance strongly depends on the SI quality. The most popular (and one of the best) existing technique is the one presented in the DISCOVER scheme. In Chapter 6, we propose several new methods which aim at improving the side information quality, using DISCOVER as reference. Moreover, in Chapter 9, we propose a complementary study about the existing SI quality measures (mainly PSNR) and observe that in some situations the PSNR does not give a good estimation of the side information quality. That is why we propose several other metrics which seem to provide more reliable results than the PSNR.

1.2.2.2.f Channel decoder

The generated side information is used to calculate the *a priori* information for the channel decoder. The side information Y is considered as a noisy version of the original WZ frame X. The noise N is assumed to be additive, *i.e.*,$Y = X + N$. The side information is then used to calculate the properties of N, assumed to be Laplacian in the literature [Brites, Pereira, 2008] (and in DISCOVER). The different existing techniques for noise correlation estimation are detailed in Chapter 8. The literature seems to show that the precision of the model has an impact on the performance. In this chapter we thus also propose to use a Generalized Gaussian model to refine the correlation estimation.

After the correlation estimation, the channel decoder starts the decoding by receiving a first flow of parity bits. After decoding each packet, the decoder calculates the error probability. If this one is greater than a threshold (set to 10^{-3} in DISCOVER [Brites *et al.*, 2008]) the decoder requests more parity bits to the encoder, and this until the bit error probability becomes lower than the threshold.

1.2.2.2.g Reconstruction

After decoding all the bitplanes, the decoded bin is used to estimate the optimal dequantized coefficient value. The simplest existing method [Aaron *et al.*, 2002] consists in taking the SI value if this one is inside the decoded bin, and in taking the bin bound closest to the SI value otherwise.
In 2007, Kubasov *et al.* [Kubasov *et al.*, 2007b] proposed to use the optimal reconstruction

levels in the sense of the MSE, which come from the Laplacian correlation model. At the end, the reconstructed DCT coefficients are inverse transformed.

1.2.2.2.h The drawbacks of the backward channel

One of the major drawback of the Stanford DVC scheme is the necessity of a backward channel. Indeed, the encoder needs to wait for the decoder request to send the correct amount of parity information. This forces DVC schemes to have a real-time decoding. This real-time constraint is hardly possible, in the sense that it would imply a complexity reducing at the decoder, very high for the moment because of the iterative algorithms used in turbodecoding.
Some works in the literature have tried to get rid of this return loop. We present these methods in detail in Section 3.3.1.2. Removing the backward channel implies a significant loss in performance (around 1 dB), and it also implies to betray the distributed coding spirit by performing a non-complex comparison between the previous and next key frames in order to have a coarse estimation of the correlation at the encoder. In Section 3.3, we propose our own encoder rate estimation method, based on the proposed rate-distortion model introduced in Chapter 2.

1.2.2.2.i Hash-based schemes

Another drawback of Stanford scheme is that, in some cases, the decoder cannot find in the K frames the information necessary for the WZ estimation (*i.e.*,in case of occlusions, rapid motion, etc.). This is why some works have proposed to help the side information generation by sending some localized and well-chosen "hash" information to the decoder. In Section 4.3 we present the different existing hash-based schemes and we detail the several problems brought by such coders. Moreover, in Chapter 7, we propose a new hash-based scheme using at the decoder a fusion based on a genetic algorithm.

1.2.3 Multiview distributed video coding

Multiview or stereo distributed video coding (MVDVC) paradigm is very similar to monoview DVC one, in the sense that the general encoding/decoding process is identical. The two main differences are the frame-type distributions in the time-view space and, and thus the side information generation methods.

1.2.3.1 Schemes

In monoview DVC, the classification of the images only consists in determining the number of WZ frames in a GOP. In MVDVC, the frame classification issue is far more complex because of the numerous possible frame type distribution. In Section 3.1 we propose a review of the different existing classifications drawn in Figure 3.1. We first observe that this classification strongly impacts on the rest of the coding chain, more specifically it impacts on the number and the position of available K frames, and thus on the way of generating the side information. Moreover, we observed that the existing classification schemes have to encode a too high number of K frames (in some of them, some cameras are entirely composed by K frames). That is why, in Section 3.1.2, we propose a scheme that contains less K frames (therefore it is less complex at the encoder), and which is the

extension to multiview of a GOP of size 4 in monoview DVC. We study, based on the proposed rate-distortion model, the best WZ frame decoding order.

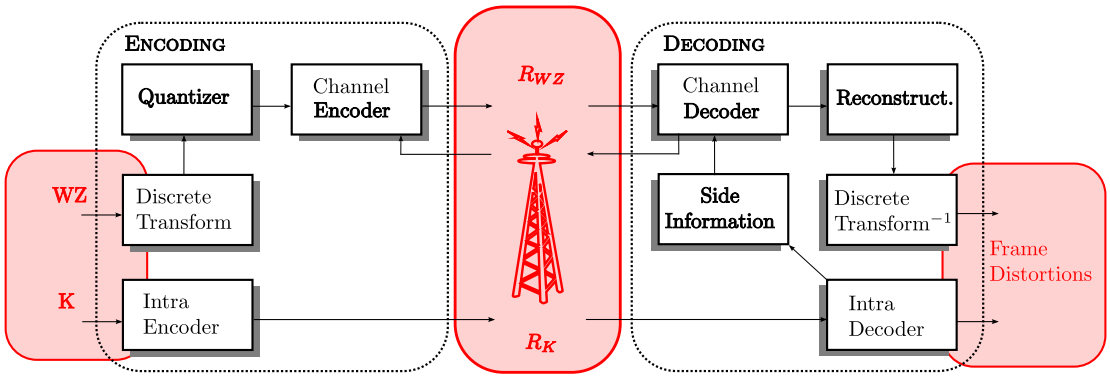
1.2.3.2 Side information

As it was mentioned above, the multicamera configuration has an impact on the way of generating the side information (a detailed state-of-the-art is given in Chapter 4). More precisely, while the temporal estimation remains similar to monoview schemes, the inter-view estimation techniques are different because they exploit for most of them the geometry of the scene (whereas some schemes still use temporal methods for inter-view estimations). Moreover, multiview configuration implies the fact that several estimations are available in order to build a unique side information. This raises the issue of how to merge all this available information in order to build a good SI (the main existing fusion techniques are detailed in Section 4.2).
In Chapter 6 we present the different methods proposed to tackle these two issues brought by the multiview configuration. We first propose several pixel-precision interpolation methods that we test for temporal and inter-view estimations, and we propose several efficient fusion methods.

1.3 Conclusion

Distributed video coding is a very surprising paradigm. In spite of the fact that it is relatively recent, much work has been done to try to achieve the theoretical rate-distortion performance. However, wheras it is very promising (theoretically), the actual performances are quite disappointing, since they are far from inter-frame video coding ones.

However, as it was explained in this chapter, the Stanford architecture presents a certain number of modules which are perfectible: the frame classification in multiview coding, the necessity of a backward channel, the side information generation, the fusion of temporal and inter-view estimation, hash-based schemes, correlation noise estimation. For all of these topics we present our contributions, in the next chapters. Some of them aim at obtaining a better understanding of the codec behaviour (with the proposal of a rate-distortion model, and the proposal of new SI quality metrics), while the other aim at improving the general performance of the coder.



Part I

Rate distortion model and applications

"Understand and model the DVC scheme behavior."

Chapter 2

Rate distortion model for the prediction error

Knowing that the Wyner-Ziv decoding efficiency strongly depends on the side information quality, it is worth finding an expression for the error between the original Wyner-Ziv frame and its estimation. In this chapter we propose an original model for the distortion of this error which presents some advantages, such as the fact that it separates the error coming from quantization and the error coming from motion/disparity interpolation. Afterwards, a discussion including experiments about the hypotheses behind this model is proposed.

Contents

2.1 Context **64**
2.2 Hypotheses and calculation **65**
2.3 Model validation **67**
2.3.1 Approximation for quantization distortion 67
2.3.2 Decorrelation between the quantization and the motion/disparity estimation errors 69
2.3.3 M_{d_1,d_2} does not depend on the quantization level 70
2.3.4 Discussion about hypothesis validation 71
2.4 Rate distortion model **73**
2.4.1 Results from information theory 73
2.4.2 Proposed model 75
2.5 Conclusion **75**

2.1 Context

In this chapter, we aim at modelling the error between the original Wyner-Ziv frame and the side information. First we need to define how the side information is generated. This is shown in Figure 2.1 and is detailed below.

The Wyner-Ziv frame is denoted by I. The two[1] reference frames used to estimate I are denoted by I_1 and I_2. The two reference frames can be either the previous and next frames in monoview or neighbour views in multiview framework. At the encoder side, the I_1 and I_2 frames are quantized. The resulting frames, denoted by $\widetilde{I}_1$ and $\widetilde{I}_2$, are transmitted. The previous operation corresponds to the intra coding of the key frames which is simplified and seen here as a single quantization block followed by a lossless transmission.

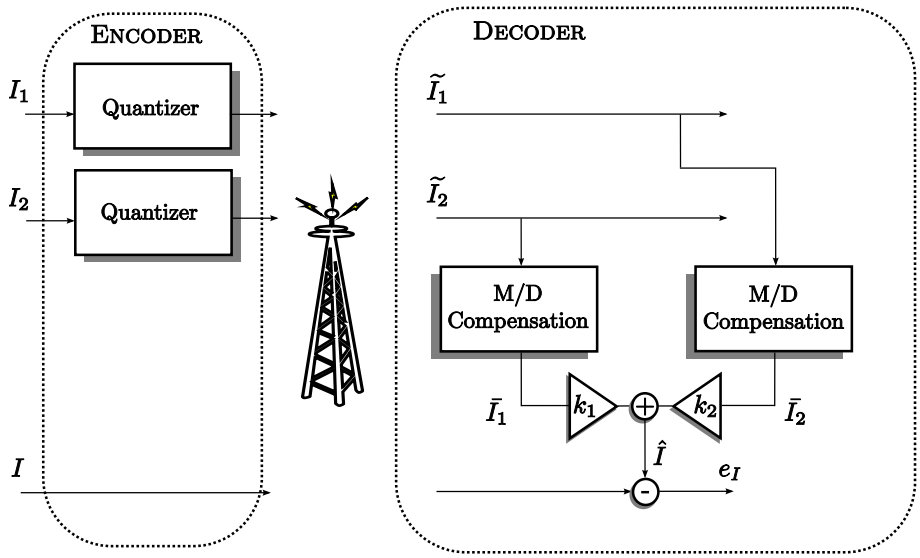


Figure 2.1: Context of the proposed distortion model.

In the following, we consider that the side information is built with a motion/disparity compensation of the quantized reference frames, as it is done in practice. The SI construction process is summed up in Figure 2.2. Vector estimation can be based either on motion or disparity interpolation. In both cases, all of the equations given in the following hold. The compensated frames are denoted by $\bar{I}_1$ and $\bar{I}_2$ and they are computed as follows. If N_{width} and N_{height} are respectively the width and height of the images, and if $\mathbf{p} \in [\![1, N_{\text{height}}]\!] \times [\![1, N_{\text{width}}]\!]$ represents the coordinates of a pixel, we denote by $\mathbf{u}_1(\mathbf{p})$ and $\mathbf{u}_2(\mathbf{p})$ the two motion/disparity vectors associated to I_1 and I_2 at $\mathbf{p}$. Then, the compensated frames read:

$$\bar{I}_1(\mathbf{p}) = \widetilde{I}_1(\mathbf{p} - \mathbf{u}_1(\mathbf{p})) \text{ and } \bar{I}_2(\mathbf{p}) = \widetilde{I}_2(\mathbf{p} - \mathbf{u}_2(\mathbf{p})). \tag{2.1}$$

The side information is considered as the linear combination between the two compensated frames $\bar{I}_1$ and $\bar{I}_2$ (like it is classically done in the DVC coder). The coefficients of this linear combination depend on the distances between I and I_1 and between I and I_2. The distance between two frames is the number of images between them plus one. For example, the

[1]For the moment we suppose that the side information is generated with only two reference frames, an extension to the more general case of n reference frame is proposed in Equation (2.9).

distance between two consecutive frames is 1. It is accepted [Ascenso *et al.*, 2006] that a reference frame far from the Wyner-Ziv frame has less influence than a closer image for the motion/disparity compensation. This idea leads us to the following intuitive statement which corresponds to the common way of building the side information: if d_1 (respectively d_2) is the distance between I_1 (resp. I_2) and I, the corresponding coefficient, k_1 (resp. k_2) of the linear combination is given by

$$k_1 = \frac{d_2}{d_1 + d_2} \qquad (\text{ resp. } k_2 = \frac{d_1}{d_1 + d_2}\ ). \tag{2.2}$$

The expression of the side information, $\hat{I}$, is then

$$\forall \mathbf{p} \in [\![1, N_{\text{height}}]\!] \times [\![1, N_{\text{width}}]\!], \quad \hat{I}(\mathbf{p}) = k_1 \bar{I}_1(\mathbf{p}) + k_2 \bar{I}_2(\mathbf{p}). \tag{2.3}$$

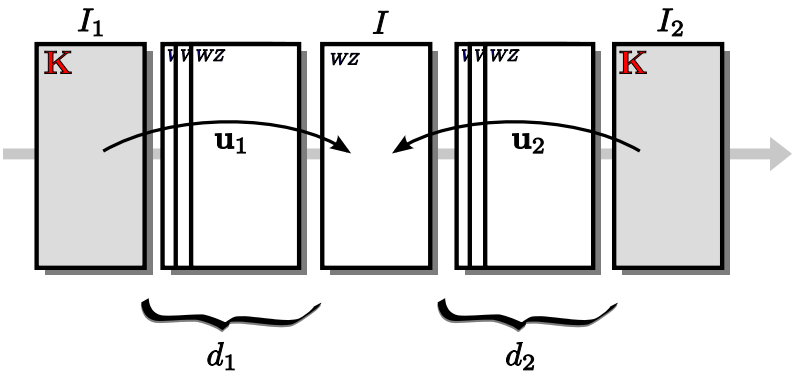


Figure 2.2: Side information construction of the Wyner-Ziv frame I using the references frames I_1 and I_2 (or their quantized version) at a respective distance of d_1 and d_2 and compensated with the fields $\mathbf{u}_1$ and $\mathbf{u}_2$.

With the side information defined, we are now able to introduce the prediction error e_I, given by the expression

$$\forall \mathbf{p} \in [\![1, N_{\text{height}}]\!] \times [\![1, N_{\text{width}}]\!], \quad e_I(\mathbf{p}) = I(\mathbf{p}) - \hat{I}(\mathbf{p}). \tag{2.4}$$

The purpose of the following section is to model this error, as it plays a very important role in the decoding performances. More precisely, we propose an expression of its variance, since the channel decoding efficiency is strongly correlated with the amplitude of the variance of the error, e_I [Aaron, Girod, 2002].

2.2 Hypotheses and calculation

In this subsection we aim at determining a simple expression for the variance of the error e_I introduced in Section 2.1. This variance has the following definition (under the hypothesis that the spatial process e_I is wide sense stationnary with $E\{e_I(\mathbf{p})\} = 0$)

$$\sigma_{e_I}^2 = E\left\{e_I(\mathbf{p})^2\right\},$$

with $\mathbf{p} \in [\![1, N_{\text{height}}]\!] \times [\![1, N_{\text{width}}]\!]$. This can thus be written as

$$\sigma_{e_I}^2 = E\left\{\left(I(\mathbf{p}) - \hat{I}(\mathbf{p})\right)^2\right\}.$$

According to Equation (2.3), the distortion is then

$$\sigma_{e_I}^2 = E\left\{\left(I(\mathbf{p}) - k_1\bar{I}_1(\mathbf{p}) - k_2\bar{I}_2(\mathbf{p})\right)^2\right\}.$$

If we take into account the vector fields, we can write

$$\sigma_{e_I}^2 = E\left\{\left(I(\mathbf{p}) - k_1\widetilde{I}_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - k_2\widetilde{I}_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p}))\right)^2\right\}. \tag{2.5}$$

We introduce two quantities to make Equation (2.5) more exploitable. These elements are $I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))$ and $I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p}))$, for $\mathbf{p} \in [\![1, N_{\text{height}}]\!] \times [\![1, N_{\text{width}}]\!]$. They are the original (non quantized) reference frames compensated with the same vector fields as those for $\widetilde{I}_1$ and $\widetilde{I}_2$. Therefore, we obtain

$$\begin{aligned}\sigma_{e_I}^2 = E\Big\{\Big(&I(\mathbf{p}) - k_1\widetilde{I}_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - k_2\widetilde{I}_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p}))\\ &+ k_1 I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - k_1 I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))\\ &+k_2 I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p})) - k_2 I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p})))^2\Big\},\end{aligned}$$

reorganized as follows:

$$\begin{aligned}\sigma_{e_I}^2 = E\Big\{\Big(&I(\mathbf{p}) - k_1 I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - k_2 I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p}))\\ &+ k_1 I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - k_1\widetilde{I}_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))\\ &+ k_2 I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p})) - k_2\widetilde{I}_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p}))\Big)^2\Big\}.\end{aligned} \tag{2.6}$$

We notice that the first line of (2.6) can be interpreted as the estimation error when the reference frames are not quantized. In other words this quantity only depends on motion activity or disparity vector field variance in the video sequence: it is assumed that it does not vary with the rate (this hypothesis is discussed in Section 2.3.3). The second and the third lines can be seen as the expression of the quantization error of the two reference frames.

Here, we make a second assumption which states that these three quantities are decorrelated. Indeed, at high bitrate, the three errors come from different physical aspects. This implies that the cross terms in (2.6) (involving different types of errors) are zero or at least negligible (Hypothesis 2 in Section 2.3.2), and then the expression of the approximated distortion, $\hat{\sigma}_{e_I}^2$ reads

$$\begin{aligned}\hat{\sigma}_{e_I}^2 = E&\left\{(I(\mathbf{p}) - k_1 I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - k_2 I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p})))^2\right\}\\ &+ k_1^2 E\left\{\left(I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - \widetilde{I}_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))\right)^2\right\}\\ &+ k_2^2 E\left\{\left(I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p})) - \widetilde{I}_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p}))\right)^2\right\}.\end{aligned} \tag{2.7}$$

The first line of (2.7) corresponds to the variance of the estimation error obtained by compensating the non quantized reference frames. It only depends on the distances d_1 and d_2 (we develop this concept in Section 2.3.3 with more details); it is denoted in the

following by M_{d_1,d_2}. The second and the third lines are the reference frame distortions due to quantization. They are denoted by D_{I_1} and D_{I_2}. In Section 2.3.1, we will discuss the approximation stating that

$$D_{I_1} = E\left\{\left(I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - \widetilde{I_1}(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))\right)^2\right\} \overset{\text{hyp}}{=} E\left\{\left(I_1(\mathbf{p}) - \widetilde{I_1}(\mathbf{p})\right)^2\right\}.$$

and the similar one derived for I_2. The expression of the distortion is then written as

$$\hat{\sigma}^2_{e_I} = M_{d_1,d_2} + k_1^2 D_{I_1} + k_2^2 D_{I_2}. \tag{2.8}$$

An interesting property of the obtained distortion formula is that the errors coming from quantization and motion/disparity interpolation are separated, which will allow for the future contributions an easier theoretical study of rate-distortion coding scheme behavior.

A last remark should be added concerning the number of reference frames. In the previous study, the distortion was expressed with only two reference images. Given N reference frames, $I_1, \ldots, I_N$, available to generate the side information, since the side information is a linear combination of the motion/disparity compensated reference frames (with the vector fields $\mathbf{u}_1, \ldots, \mathbf{u}_N$), we still consider that the coefficients $k_1, \ldots, k_N$ depend on the distances $d_1, \ldots, d_N$, and then, under similar hypotheses as before, we can obtain a more general expression:

$$\hat{\sigma}^2_{e_I} = M_{d_1,\ldots,d_N} + k_1^2 D_{I_1} + \ldots + k_N^2 D_{I_N} \quad \text{with} \quad \forall i \in [1,N] \;\; k_i = \frac{1}{N-1}\frac{\sum_{j=0,j\neq i}^{N} d_j}{\sum_{j=0}^{N} d_j}. \tag{2.9}$$

In the next subsections we shall tests the reliability of the different underlying hypotheses of this model.

2.3 Model validation

2.3.1 Approximation for quantization distortion

Hypothesis 1 *The term* $E\left\{\left(I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - \widetilde{I_1}(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))\right)^2\right\}$ *can be approximated by* $E\left\{\left(I_1(\mathbf{p}) - \widetilde{I_1}(\mathbf{p})\right)^2\right\}$ *and then can be assimilated to the quantization error of the reference frame* I_1*. An equivalent hypothesis can be formulated for* I_2*.*

Hypothesis 1 formulates the assumption that the error between the compensated reference frame and the compensated quantized reference frame can be assimilated to the simple quantization error of the reference image. In order to test the validity of this hypothesis, a set of experiments was performed. For several video sequences and for several quantization steps, both distortions, $E\left\{\left(I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - \widetilde{I_1}(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))\right)^2\right\}$ and $E\left\{\left(I_1(\mathbf{p}) - \widetilde{I_1}(\mathbf{p})\right)^2\right\}$, have been measured. The difference between them has been calculated and then normalized with respect to the value of the quantization error of the reference frame. The resulting statistic is then a percentage of errors between the two entities. Results are displayed in Table 2.1 and indicate that the two distortions are very

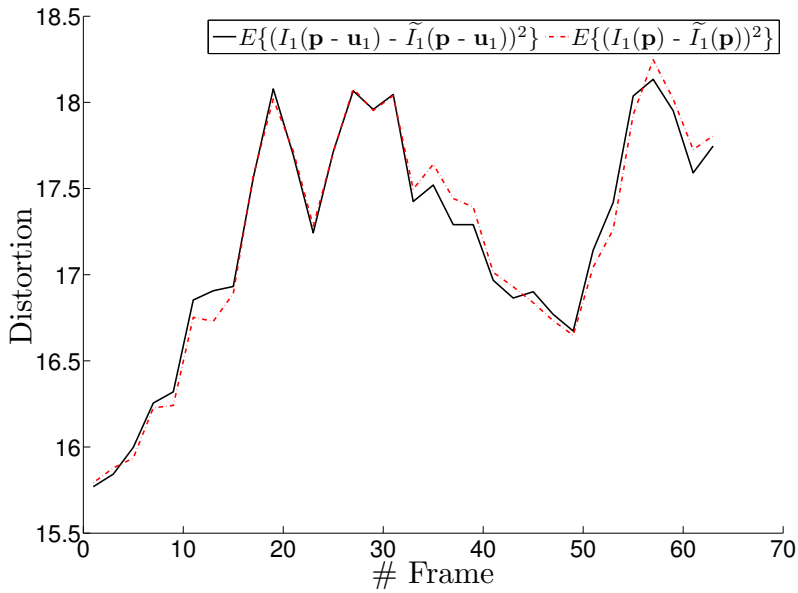


Figure 2.3: Evolution of the distortions measured on *foreman* sequence at a QP=31. In full black line, the difference between compensated original and quantized reference frames, in red dashed line, the quantization error of the reference frame.

similar. Indeed the error between them is never greater than 1.51%. The two plots, displayed in Figures 2.3 and 2.4, show the behavior of the two distortions along time for two sequences (*foreman* and *mobile*) and two quantization steps (QP 31 and 40). Though the difference between the two distortions is more sensible at low bitrate (QP 40), it still remains very similar, confirming that Hypothesis 1 is reasonable.

QP	31	34	37	40
eric	0.38	0.47	0.47	0.49
foreman	0.38	0.33	0.38	0.55
football	0.87	1.01	1.35	1.51
soccer	0.26	0.46	0.49	0.75
mobile	0.10	0.10	0.10	0.14
Average	0.33	0.40	0.47	0.58

Table 2.1: Per cent error between the two quantities $E\left\{\left(I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))-\widetilde{I}_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))\right)^2\right\}$ and $E\left\{\left(I_1(\mathbf{p})-\widetilde{I}_1(\mathbf{p})\right)^2\right\}$ for 6 video sequences (176×144, 60 frames) and 4 quantization parameters (QP) for the key frames.

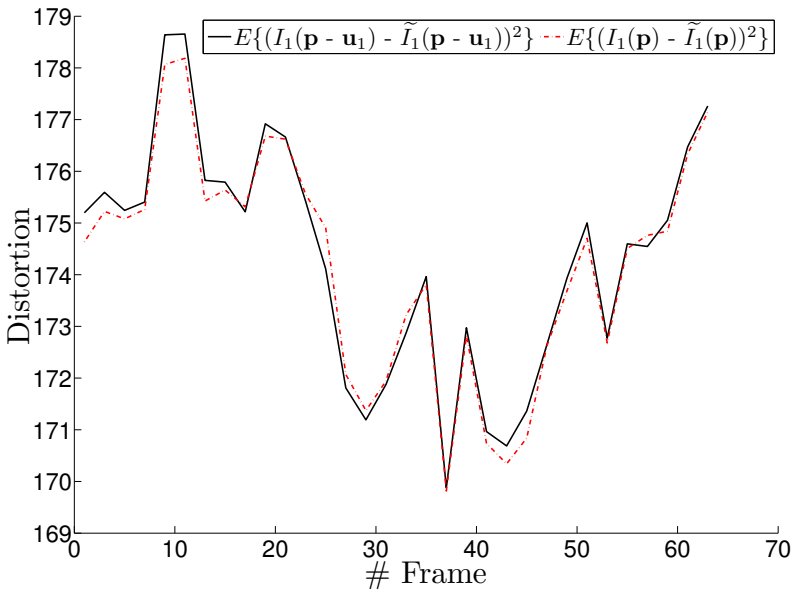


Figure 2.4: Evolution of the distortions measured on *mobile* sequence at a QP=40. In full black line, the difference between compensated original and quantized reference frames, in red dashed line, the quantization error of the reference frame.

QP	31	34	37	40
eric	4.28	7.44	11.63	16.87
foreman	3.26	3.90	6.88	10.75
football	3.31	4.99	7.17	10.71
soccer	2.74	3.84	5.83	7.42
mobile	4.88	8.08	11.76	17.42
Average	7.48	10.07	13.20	16.90

Table 2.2: Per cent error between the two quantities $\sigma^2_{e_I}$ and $\hat{\sigma}^2_{e_I}$ for 6 video sequences (176×144, 60 frames) and 4 quantization parameters (QP) for the key frames.

2.3.2 Decorrelation between the quantization and the motion/disparity estimation errors

Hypothesis 2 *The three following cross correlation terms are considered as negligible compared to M_{d_1,d_2}, $k_1^2 D_{I_1}$ and $k_2^2 D_{I_2}$:*

$$\sigma_{e_{I_1},e_{I_2}} = k_1 k_2 E\left\{\left(I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - \widetilde{I}_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))\right)\left(I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p})) - \widetilde{I}_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p}))\right)\right\}$$
$$\sigma_{e_I,e_{I_1}} = k_1 E\left\{(I(\mathbf{p}) - k_1 I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - k_2 I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p})))\left(I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - \widetilde{I}_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p}))\right)\right\}$$
$$\sigma_{e_I,e_{I_2}} = k_2 E\left\{(I(\mathbf{p}) - k_1 I_1(\mathbf{p}-\mathbf{u}_1(\mathbf{p})) - k_2 I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p})))\left(I_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p})) - \widetilde{I}_2(\mathbf{p}-\mathbf{u}_2(\mathbf{p}))\right)\right\}$$

This is the key assumption of our model. Indeed, thanks to it we are able to write an expression of the distortion which separates the motion/disparity estimation error and the quantization error allowing simpler rate distortion analyses of the coding scheme.

QP	31	34	37	40
eric	5.82	9.42	14.07	28.59
foreman	16.31	11.79	27.61	17.81
football	1.84	3.03	4.77	6.64
soccer	2.73	9.69	9.45	11.53
mobile	3.20	4.41	6.66	12.53
Average	5.98	7.67	12.51	15.42

Table 2.3: Per cent error between the temporal numerical derivatives of $\sigma^2_{e_I}$ and $\hat{\sigma}^2_{e_I}$ for 6 video sequences (176×144, 60 frames) and 4 quantization parameters (QP) for the key frames.

As in Section 2.3.1, several experiments have been run in order to check the validity of Hypothesis 2. For several sequences and for several rates (obtained by modifying the quantization step of the key frames), the real distortion, $\sigma^2_{e_I}$, is measured, and compared to the approximation $\hat{\sigma}^2_{e_I}$. We calculate the per cent error between them. The obtained results are reported in Table 2.2 and in Figures 2.5 and 2.6. While the distance is quite small (under 10%) for the major part of the statistics, there are some larger values (the maximum being 17.42% for *mobile* at low bitrates) which demonstrates that, in some cases (mainly for high QP), the approximation $\hat{\sigma}^2_{e_I}$ does not fully reflect the reality. Plots in Figures 2.5 and 2.6 confirm the tendance. They show the evolution of $\sigma^2_{e_I}$ (in plain black line) and $\hat{\sigma}^2_{e_I}$ (in dashed dotted red line). The Figures also display the aspect of the quantities $\sigma_{e_{I_1},e_{I_2}}$ (dotted green line), $\sigma_{e_I,e_{I_1}}$ and $\sigma_{e_I,e_{I_2}}$ (dotted blue lines), which are supposed to be negligible compared to M_{d_1,d_2} (plain green line), $k_1^2 D_{I_1}$ and $k_2^2 D_{I_2}$ (plain blue lines).

In Figure 2.5 which displays results obtained at high bitrate, the approximation $\hat{\sigma}^2_{e_I}$ is very similar to the original distortion $\sigma^2_{e_I,e_{I_1}}$. At low bitrate (Figure 2.6), the approximation error is wider and confirms the bad results in Table 2.2. In a rate allocation/estimation framework, the crux of the matter is to approximate the evolution of the distortion along time. To this end, we do not need access to the exact distortion value. In this light, and since the gap between the true and the estimated distortion remains unchanged, the obtained results are adequate to the rate allocation/estimation problem and thus can be deemed as satisfying. Then, we have calculated the numerical temporal differential of the distortions and we have measured the difference (in %) between them. The obtained results, in Table 2.3, seem disappointing, but it is known that the differential is more sensible to errors. For example, the plots in Figure 2.5 have very close evolutions, but the differential error is about 16%. In this light, the results in Table 2.3 are quite good, and show that even if at low bitrate there is a gap between $\sigma^2_{e_I}$ and $\hat{\sigma}^2_{e_I}$, it remains constant along the sequence. The $\hat{\sigma}^2_{e_I}$ thus at least predicts reliably the evolution of the original distortion $\sigma^2_{e_I}$ and at high bitrate, predicts its almost exact value. To conclude, in the light of these acceptable results, the proposed distortion model seems to be suited to the aimed applications.

2.3.3 M_{d_1,d_2} does not depend on the quantization level

Hypothesis 3 $M_{d_1,d_2} = E\left\{(I(\mathbf{p}) - k_1 I_1(\mathbf{p} - \mathbf{u}_1(\mathbf{p})) - k_2 I_2(\mathbf{p} - \mathbf{u}_2(\mathbf{p})))^2\right\}$ *does not depend on quantization level of the the key frame.*

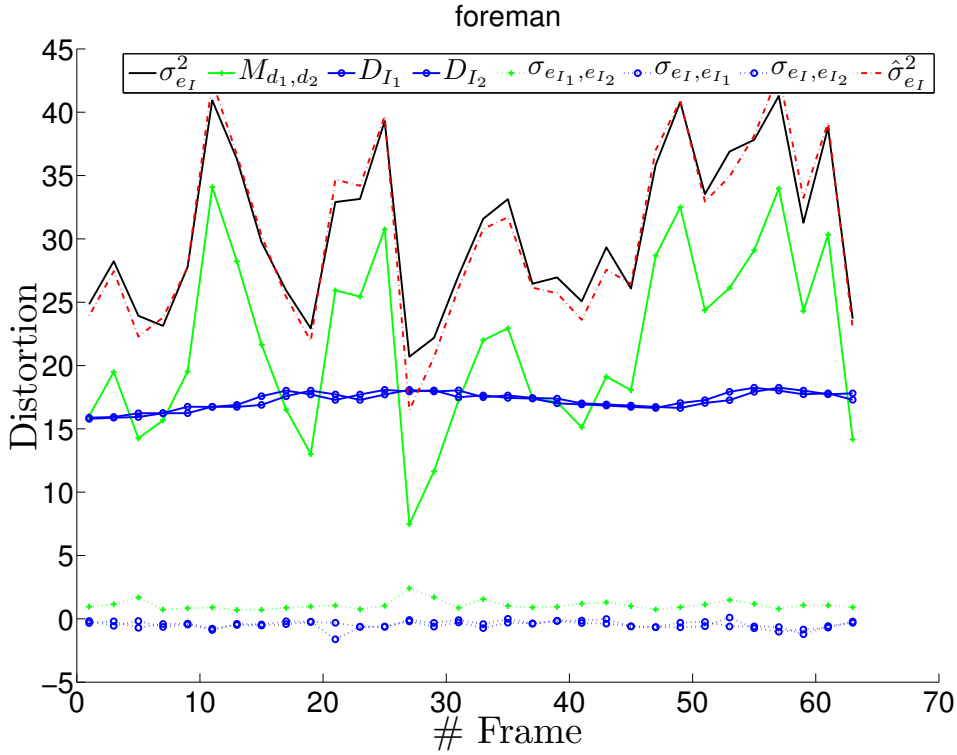


Figure 2.5: Evolution of the distortions measured on *foreman* sequence at a QP=31.

The term M_{d_1,d_2} has been introduced to highlight the separation of the quantization error and the motion/disparity estimation error. Though in the definition the quantized reference frames have been avoided and replaced by the original motion compensated ones, there still remains a little dependency to the reference image quantization through the motion/disparity vector fields, $\mathbf{u}_1$ and $\mathbf{u}_2$. Indeed, they have been calculated between the two quantized versions of the reference frames, and therefore, depend on the QP. In this subsection, some experiments have been run in order to measure the influence of the quantization on M_{d_1,d_2}. For several sequences we have calculated the statistics presented in Table 2.4. They correspond to the average error in (%) between the mean value M_{d_1,d_2} calculated with motion/disparity estimated at four QP (31,34,37 and 40). The obtained results show that M_{d_1,d_2} obviously depends on the QP but not so much, and we can assume that it is independent from the reference image quantization.

2.3.4 Discussion about hypothesis validation

Looking at all the results in Section 2.3.1, 2.3.2 and 2.3.3, several conclusions can be drawn.

- For Hypothesis 2, the term mainly responsible of the sensible gap between $\sigma^2_{e_I}$ and $\hat{\sigma}^2_{e_I}$ is $\sigma_{e_{I_1},e_{I_2}}$. Indeed, the fact that it becomes non zero easily can be explained, precisely when there are only few differences between I_1 and I_2, *i.e.*, in case of low motion or similar texture. The two other terms, $\sigma_{e_I,e_{I_1}}$ and $\sigma_{e_I,e_{I_2}}$, are nearly always very small. For example for *mobile*, the gap at low bitrate is 17.42%, which is explained by the fact that the texture is very similar from one frame to another in this sequence,

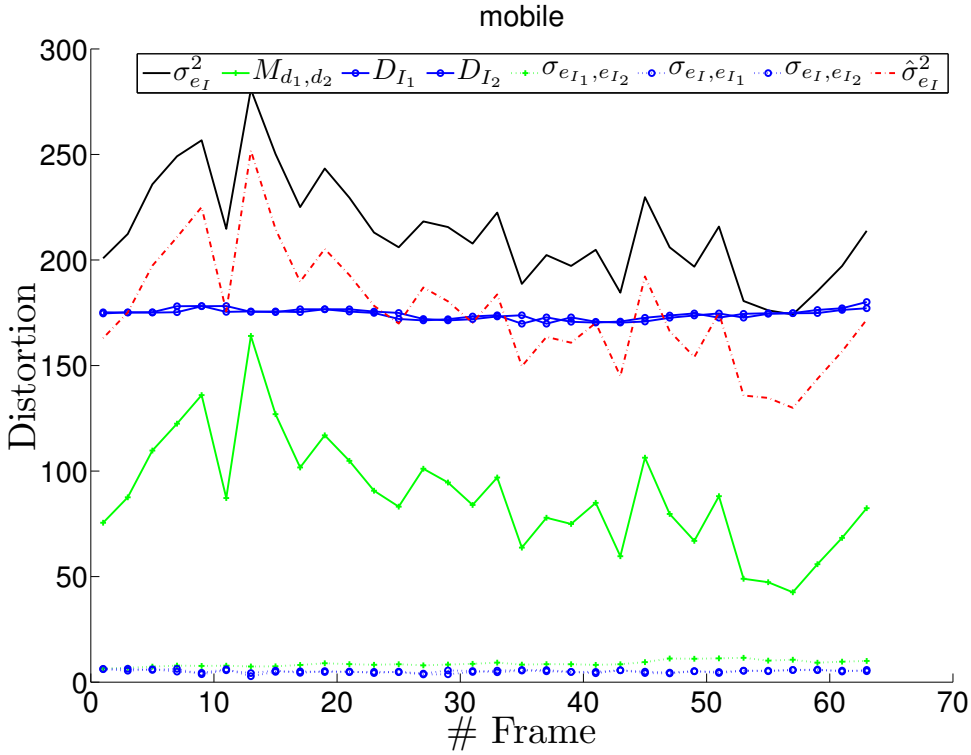


Figure 2.6: Evolution of the distortions measured on *mobile* sequence at a QP=40.

eric	*foreman*	*football*	*soccer*	*mobile*
4.30	7.92	1.31	10.58	6.70

Table 2.4: Average error in (%) between the mean value of M_{d_1,d_2} calculated at four QPs (31,34,37,40).

contrary to *soccer* sequence which is more complex, and has thus led to similarity between $\sigma^2_{e_I}$ and $\hat{\sigma}^2_{e_I}$.

- Hypothesis 3 seems to be quite well verified for *soccer*. The explication is certainly more complex than for Hypothesis 2, but we can guess that the texture of the images and their resistance against compression are important elements for the validity of Hypothesis 3.

- At the end, the proposed model is acceptable. While the simplifications made may lead to a gap between the model and the true distortion at low bitrate, the evolution of $\hat{\sigma}^2_{e_I}$ is always well predicted, which is very significant for many applications. Moreover, the simple expression of the model (separation of the quantization and motion estimation errors) allows a very easy rate-distortion analysis along the GOP, as we will see in the next section.

2.4 Rate distortion model

2.4.1 Results from information theory

In this section, we recall some classical results of information theory, which are presented in more details in [Berger, 1971; Cover, Thomas, 2006]. If χ is a probability space, we study here the coding of a random variable, $X \in \chi$, and its reconstruction (denoted by $\hat{X} \in \chi$). The purpose is to study the rate distortion characteristics depending on the probabilist properties of the source. This one generates sets of n elements $\mathbf{X} = X_1, X_2, \ldots, X_n$, iid and following the probability density p. These n-symbols are described with an index $f_n(\mathbf{X}) \in \{1, 2, \ldots, 2^{nR}\}$ (where R is the transmission rate per element). At the decoder, an estimation of $\mathbf{X}$ is associated to this index. This estimation is called the reconstruction and is denoted by $\hat{\mathbf{X}} \in \chi^n$.

We recall that the *distortion* is defined as a function d :

$$\chi^2 \to \mathbb{R}^+, \quad (x, \hat{x}) \mapsto d(x, \hat{x})$$

which gives the cost of representing $x \in \chi$ by $\hat{x} \in \chi$. There exists many distortion functions. Two of them are well known and often used :

$$d(x, \hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{if } x \neq \hat{x} \end{cases} \quad \text{(Hamming)} \tag{2.10}$$

$$d(x, \hat{x}) = (x - \hat{x})^2 \quad \text{(square-error)} \tag{2.11}$$

The distortion between two n-sequence, $\mathbf{x} = (x_1, \ldots, x_n)$ and $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_n)$, is then defined as

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i)$$

.

Then, we introduce the definition of a $\left(2^{nR}, n\right)$ *rate distortion code* as a encoding function

$$f_n : \chi^n \to \{1, \ldots, 2^{nR}\},$$

and a decoding function

$$g_n : \{1, \ldots, 2^{nR}\} \to \chi^n.$$

The associated distortion is

$$D = E\left\{d(\mathbf{X}, g_n(f_n(\mathbf{X})))\right\}.$$

From this definition we can introduce the following notion: a rate distortion pair (R, D) is *achievable* if there exists a sequence of $\left(2^{nR}, n\right)$ rate distortion codes such that

$$\lim_{n \to \infty} E\left\{d(\mathbf{X}, g_n(f_n(\mathbf{X})))\right\} \leq D.$$

An important theorem of rate distortion theory states that the rate distortion function for a source X with a bounded distortion function d is:

$$R(D) = \min_{p(x,\hat{x}) : \sum_{(x,\hat{x}} p(x)p(x|\hat{x})d(x,\hat{x})} I(X; \hat{X}) \tag{2.12}$$

It can be extended to well-behaved continuous sources with unbounded distortion measures.

Let us consider the case of a square error distortion (Equation (2.11)). It is proven [Gray, 1990], that the Shannon lower bound rate can be written as (for a continuous source under a distribution p):

$$R(D) = h(p) - \frac{1}{2}\log_2(2\pi e D) \tag{2.13}$$

Now, let us study the particular case of video coding. A natural frame distribution is sometimes assumed to be Gaussian, and an error image distribution is usually considered as Laplacian. Then, let us develop Equation (2.13) in case of Generalized Gaussian distributions. For two strictly positive real numbers α and β, the Generalized Gaussian pdf is defined as:

$$f_{GG}(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-\left(\frac{|x|}{\alpha}\right)^{\beta}}$$

where $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ is the classical "gamma" function. The coefficient β impacts on the general shape of the distribution, and α gives the scale. The variance of the Generalized Gaussian law is:

$$\sigma_{GG}^2 = \alpha^2 \frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}.$$

It is also known that the expression of the entropy is [Nadarajah, 2005]:

$$h_{GG}(p) = \frac{1}{\beta} - \log_2\left(\frac{\beta}{2\alpha\Gamma(1/\beta)}\right).$$

If we express the entropy as a function of the variance, we obtain:

$$h_{GG}(p) = \frac{1}{2}\log_2\left(\underbrace{\left(2e^{1/\beta}\frac{\Gamma(1/\beta)}{\beta}\right)^2 \frac{\Gamma(1/\beta)}{\Gamma(3/\beta)}}_{g(\beta)} \sigma_{GG}^2\right).$$

We notice that $g(\beta)$ only depends on the general shape of the source distribution. Let us write now the corresponding rate-distortion function (with Equation (2.13)):

$$R(D) = \frac{1}{2}\log_2\left(g(\beta)\sigma_{GG}^2\right) - \frac{1}{2}\log_2(2\pi e D)$$

$$R(D) = \frac{1}{2}\log_2\left(\frac{g(\beta)}{2\pi e}\frac{\sigma_{GG}^2}{D}\right) \tag{2.14}$$

which can be inversed and written in the following distortion-rate form:

$$D(R) = \underbrace{\frac{2\pi e}{g(\beta)}}_{\mu} \sigma_{GG}^2 2^{-2R}$$

$$D(R) = \mu\sigma_{GG}^2 2^{-2R} \tag{2.15}$$

where μ depends on the distribution.

2.4.2 Proposed model

Based on Equation (2.15) we are able to write a rate distortion function for a frame I_{ref} (at high bitrates):

$$D_{I_{\text{ref}}} = \mu_{I_{\text{ref}}} \sigma^2_{I_{\text{ref}}} 2^{-2R_{I_{\text{ref}}}}. \tag{2.16}$$

If I_{ref} is a reference frame, $\sigma^2_{I_{\text{ref}}}$ corresponds to its variance. But if I_{ref} is a reconstructed Wyner-Ziv frame estimated thanks to two reference frames I_1 and I_2, then $\sigma^2_{I_{\text{ref}}}$ corresponds in fact to the error variance $\sigma^2_{e_{I_{\text{ref}}}}$.
Then, using the proposed model for the distortion expression, we obtain the following rate distortion function for a Wyner-Ziv frame (thanks to Equation (2.8)):

$$D_{I_{\text{ref}}} = \mu_{I_{\text{ref}}} \left(M_{d_1,d_2} + k_1^2 D_{I_1} + k_2^2 D_{I_2} \right) 2^{-2R_{I_{\text{ref}}}}. \tag{2.17}$$

Recursive analysis : based on this simple model structure, it is simple to make a recursive analysis in case of WZ frames generated thanks to other WZ frames. For example, if we assume in the previous equation that I_1 was generated thanks to I'_1 and I'_2, one can easily write:

$$D_I = \mu_{I_{\text{ref}}} \left(M_{d_1,d_2} + k_1^2 \left(\mu_{I_1} \left(M_{d'_1,d'_2} + k_1'^2 D_{I'_1} + k_2'^2 D_{I'_2} \right) 2^{-2R_{I_1}} \right) + k_2^2 D_{I_2} \right) 2^{-2R_{I_{\text{ref}}}}.$$

This idea leads us to make several works based on this model, which are presented in Chapter 3.

2.5 Conclusion

In this chapter, we proposed a distortion model for the WZ frame estimation. Based on several hypotheses, this model manage to give a good description of the true distortion, or at least its evolution along the time. Thanks to the simple model structure, we are now able to write the rate-distortion function expression of a WZ frame, in function of the rates of the key frames and other WZ frames. The next chapter uses these properties to model the coder behaviour.

Chapter 3

Applications of the rate-distortion model

The characteristics of the WZ frame estimation distortion model introduced in the previous chapter are twofolds. First of all, its structure is simple since it separates the error coming from the motion/disparity error and the error due to the reference frame quantization. Secondly, the distortion model gives a good estimation of the evolution of the true distortion, which can be interesting for several applications, such as the rate estimation at the encoder. Based on these ideas, we propose here to use them in three important problems in DVC.
Firstly, in Section 3.1 we study the frame type repartition at the encoder input of a multiview scheme, we propose a novel and more efficient frame classification, and we use the model for establishing the optimal WZ frame decoding order.
Moreover, in Section 3.2, we investigate the codec behavior in case of frame loss. Based on the proposed distortion expression, we aim at modelling the error propagation and thus the influence of the WZ frame position in the GOP. Finally, we use the model for rate estimation at the encoder and thus to propose an algorithm allowing to get rid of the feedback channel, which will be presented in Section 3.3.

Contents

3.1 Multiview schemes 78
3.1.1 State-of-the-art 78
3.1.2 Symmetric schemes 80
3.1.3 Experimental validation 84
3.2 Frame loss analysis 88
3.2.1 Context 88
3.2.2 Theoretical analysis 89
3.2.3 Experimental validation 91
3.3 Backward channel suppression 92
3.3.1 Introduction 92
3.3.2 Frame rate estimation 95
3.3.3 Bitplane rate estimation 100
3.4 Conclusion 104

3.1 Multiview schemes

In a multiview distributed video coding context, image type classification is a crucial problem, because of the consequence it has on the whole rest of the coding scheme (for example, side information generation methods). Then, a distribution of the two types of frames in the time-view space (Figure 3.1 (a)), called *scheme* in the following, has to be adopted before the encoding process.

In this section, we first describe the schemes existing in the literature (Section 3.1.1), and then we propose new symmetric schemes (Section 3.1.2) for which we shall determine the best decoding strategy based on the previously introduced RD model. These methods will be validated by experimental results (Section 3.1.3). [1]

3.1.1 State-of-the-art

As we can guess from Figure 3.1 (a), many configurations of frame repartition are conceivable. Surprisingly, the existing solutions are not so numerous and can be divided in three main categories. Before presenting them, we introduce the three types of cameras used[2].

- Key cameras: all of their generated frames are key frames. They can be encoded with an Intra coder but also with an Inter coder, involving only frames from other Key cameras. Anyway, these cameras need to be more powerful since intra or inter encoding is more complex than WZ encoding.
- Wyner-Ziv cameras: all of their frames are Wyner Ziv frames. The side-information for them is built by using the KFs of the other cameras. These cameras are less demanding in terms of computational power.
- Hybrid cameras: their frames can be key and Wyner Ziv frames. The side-information is built thanks to the key frames of the other cameras and also thanks to their own key frames. The advantage of using this type of cameras is that the problem becomes symmetric, and all the cameras in the system are identical.

Using all these types of cameras, many possible settings are conceivable. In the following, we present some configurations existing in the literature. In Figure 3.1 (b) (c) and (d), the KFs are in grey and the WZFs in white. Again, three main schemes exist:

- The *asymmetric scheme* (AS): The type of cameras alternates between Key and Wyner-Ziv, as shown in Figure 3.1 (b). Then the side-information is built using the closest frames in the view direction. This principle is used for example in [Ouaret *et al.*, 2006][Artigas *et al.*, 2007b].
- The *hybrid 1/2 scheme* (Hyb2): One camera over two is a Key camera and between them, there are hybrid cameras. This scheme is illustrated in Figure 3.1 (c). In this

[1] The material in this section was published in:

- T. Maugey and B. Pesquet-Popescu, "Side information estimation and new symmetric schemes for multi-view distributed video coding," *J. on Visual Communication and Image Representation*, vol. 19, no. 8, pp. 589–599, Dec. 2008, special issue: Resource-Aware Adaptive Video Streaming.

[2] There exist many hardware classifications, the one presented here is done from the point of view of the types of frames generated with the camera.

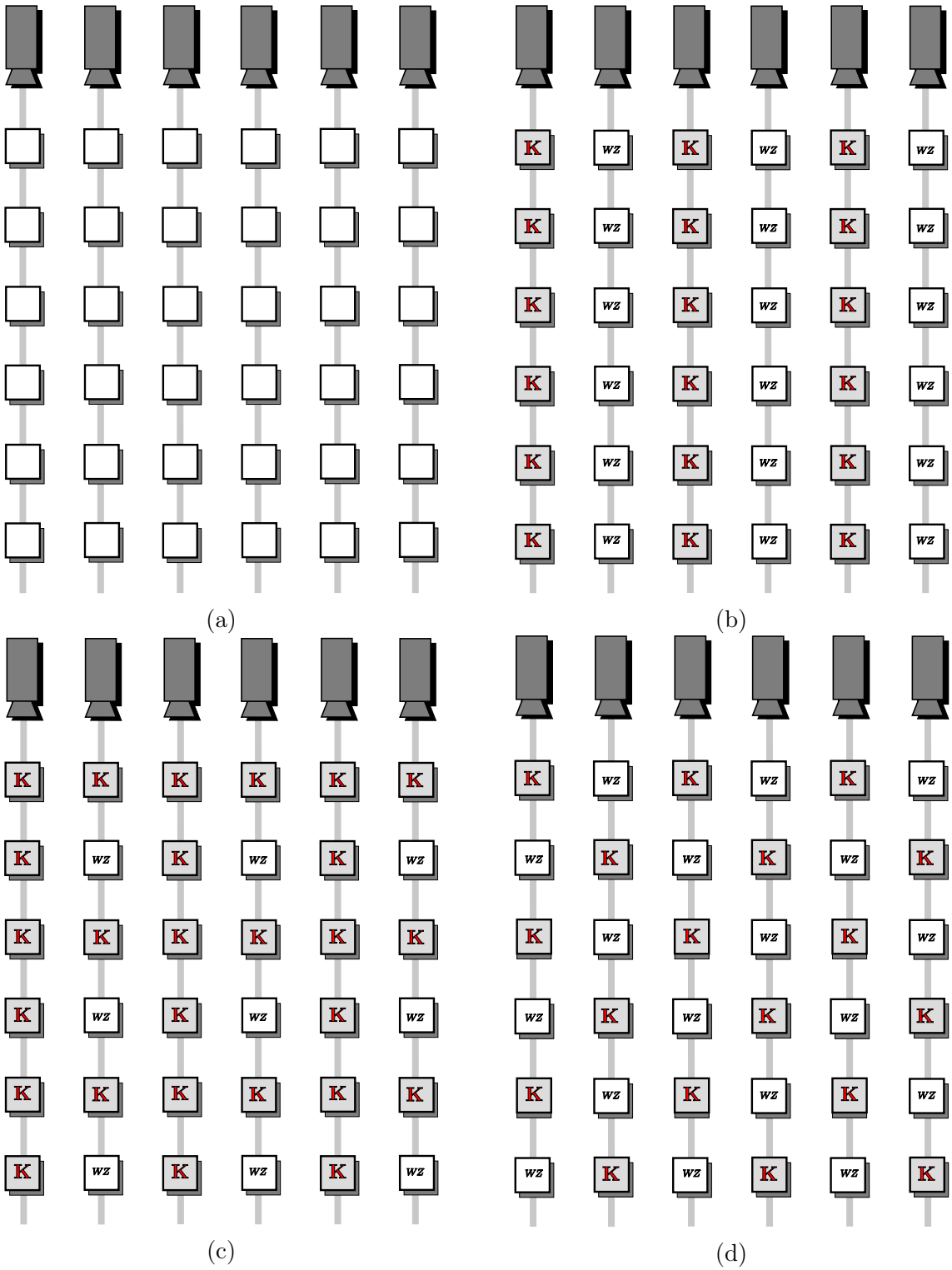


Figure 3.1: Frame disposition in the time-view space for different schemes (a) Time-view space representation (b) The asymmetric scheme (c) The hybrid 1/2 scheme (d) The symmetric 1/2 scheme.

case, the side-information can be estimated in the temporal and in the view direction leading to the necessity of performing a fusion between these two estimations. This scheme was proposed for example in [Ouaret *et al.*, 2006][Ouaret *et al.*, 2007][Ouaret *et al.*, 2009][Artigas *et al.*, 2006][Artigas *et al.*, 2007b].

- The *symmetric 1/2 scheme* (Sym2): The cameras are all hybrid with one KF for one WZF. This scheme is presented in Figure 3.1 (d). The KFs and the WZFs are placed on a quincunx grid in the time-view axes. The side-information for each WZF can be then computed in the view direction and in the time direction. This case also has to cope with the fusion problem. It was proposed in [Guo *et al.*, 2006a].

3.1.2 Symmetric schemes

Based on the analysis of the dependency between the number of estimations and the quality of the side information, we propose a new symmetric scheme. Our first goal is to preserve the symmetric nature of the schemes because asymmetric ones are too much restrictive for the camera configuration (position, power, etc.). Since in the mono-view distributed video coding the length of the GOP can be more than 2, we propose to investigate the extension of a GOP size of 4 to multiview. This is why we propose a scheme called *symmetric* 1/4 (Sym4) in Figure 3.2. This scheme, if its performance proves to be acceptable, has the advantage of being even less complex at the encoder, and this is one of the main goals of distributed coding. However, the decoder complexity is increased, since the number of WZFs which need to be channel decoded has grown.
We did not consider a scheme similar to the one used for hierarchical B frames (in multi-view source coding [ISO/IEC MPEG & ITU-T VCEG, 2007]), with I frames obtained only by a dyadic subsampling of the video sequence, since we wanted to fully exploit the correlations in both temporal and view directions for each WZF. Indeed, in the JMVM approach, the first motion/disparity compensated interpolations are done in a single direction (temporal or view).

With this new symmetric scheme, several ways of decoding are conceivable. In this section we propose a theoretical study, in order to choose the one having the best Rate-Distortion (RD) performance. Based on the recursive rate-distortion analysis introduced in Chapter 2, we will first study the mono dimensional case, and then we will extend the conclusions for multi-dimensional (temporal and view) conditions.

In one dimension, corresponding to the view or time axis in the Sym4 scheme, three decoding strategies may be envisaged, as illustrated in Figure 3.3. In the first strategy, the two WZFs closest to the KFs are first decoded and thanks to them, the SI of the middle WZF is then interpolated. In the second strategy, very similar in spirit with the "hierarchical B frames" [ISO/IEC MPEG & ITU-T VCEG, 2007], the middle WZF is first decoded and then it is used to generate the SI necessary for decoding the two other WZFs. In the third strategy, all the WZFs are simultaneously decoded, thanks to the SI generated from the two KFs.

In order to choose the best decoding strategy, let us study the theoretical dependencies between frames in the three situations. Based on the RD model introduced in Chapter 2, and with the notations in Figure 3.3, let us calculate the RD function for each of the three

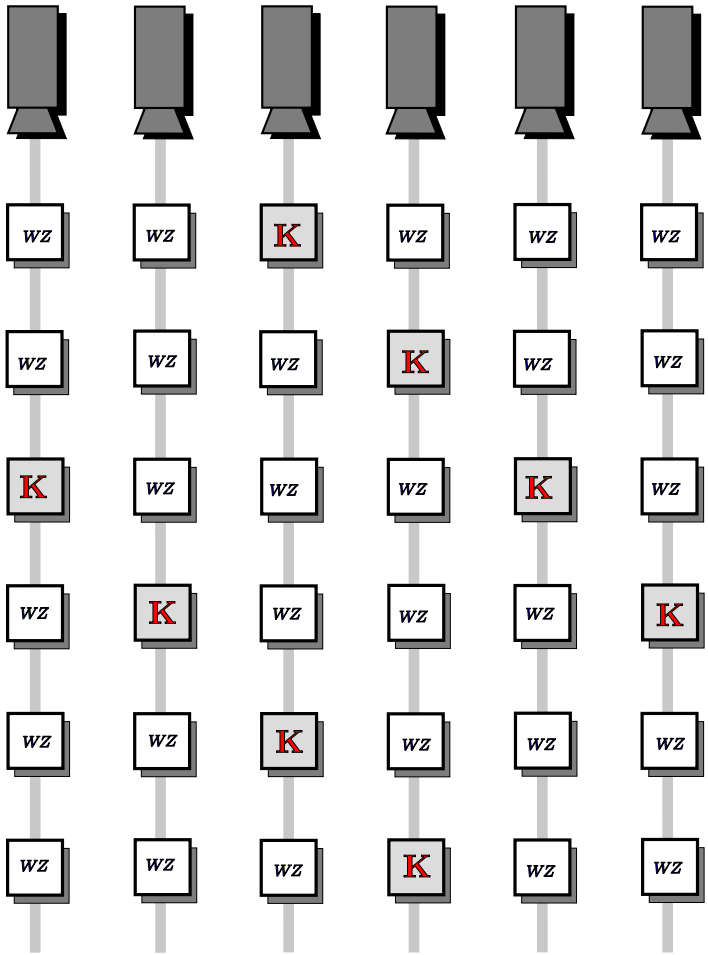


Figure 3.2: Symmetric 1/4 scheme (Sym4). KF are in grey, WZF in white.

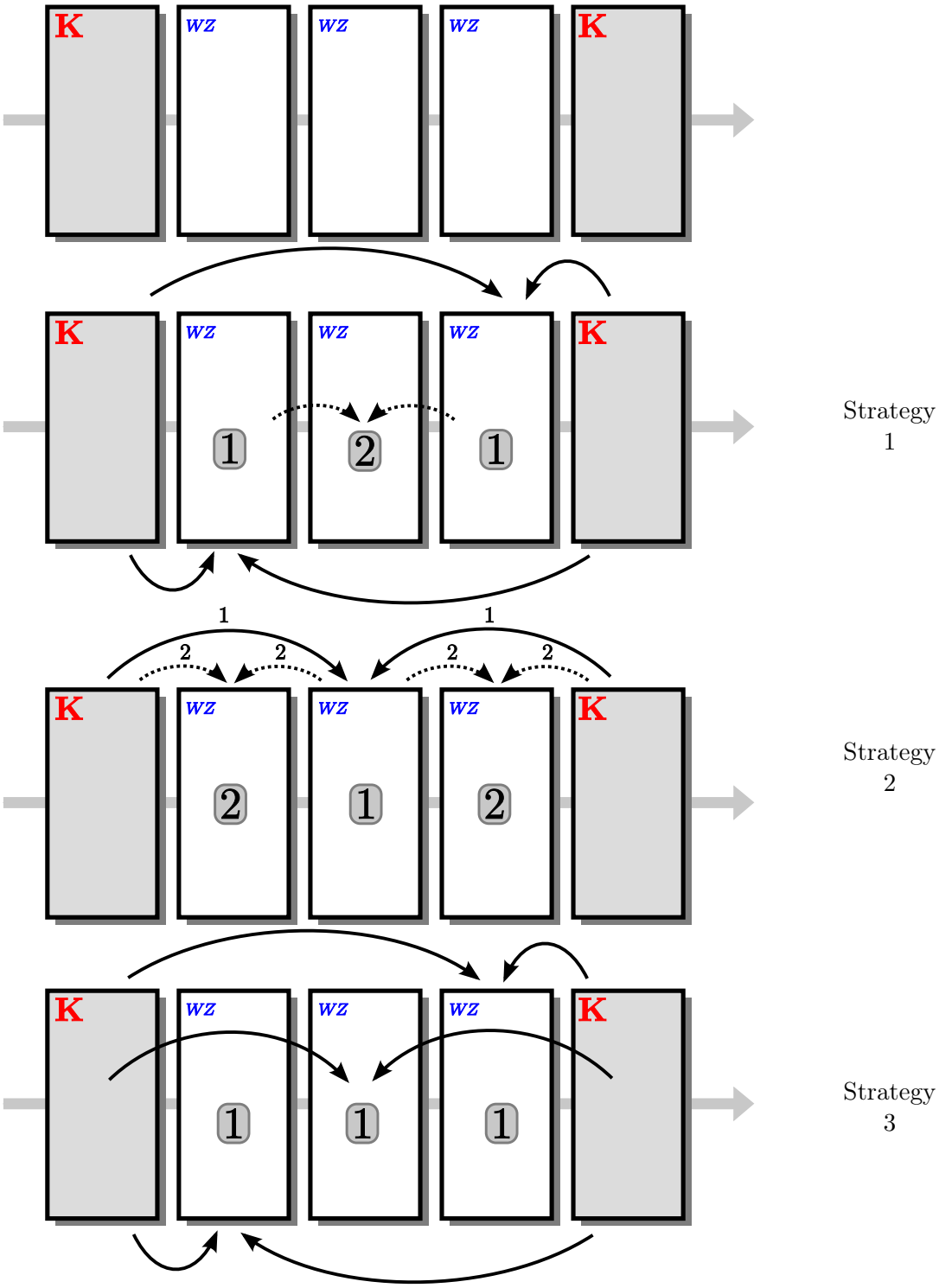


Figure 3.3: Three decoding strategies for Sym4. The numbers indicate the temporal order of estimating the SI for the different WZFs.

strategies, and compare them. We call the middle WZF, W_m, and the two others are called lateral frames, W_l. We do not make the difference between the two W_l, because the three decoding strategies give an identical role to both lateral WZFs. Denoting by D_l and D_m (resp. by R_l and R_m) the variances of the estimation errors (resp. the rates) of the frames W_l and W_m, let us calculate the total distortion: $D = 2D_l + D_m$. We denote by D_K the distortion of a KF (supposed here to be equivalent for all the KFs). The equations in the following are written under high hypothesis assumption.

- **Strategy 1:** Following the temporal WZ decoding order of "strategy 1" in Figure 3.3, we can first write the distortion of the lateral frames generated by two KFs at a distance of 1 and 3 (the two coefficients of the linear combination are thus $\frac{3}{4}$ and $\frac{1}{4}$). Equation (2.8) leads to:

$$\begin{aligned} D_l = \mu\sigma_l^2 2^{-2R_l} &= \mu\left(M_{1,3} + \left(\frac{3}{4}\right)^2 D_K + \left(\frac{1}{4}\right)^2 D_K\right) 2^{-2R_l} \\ &= \mu\sigma_l^2 2^{-2R_l} = \mu\left(M_{1,3} + \frac{5}{8}D_K\right) 2^{-2R_l} \end{aligned}$$

  The distortion of the middle frame, after reconstructing the lateral WZFs, is:

$$\begin{aligned} D_m = \mu\sigma_m^2 2^{-2R_m} &= \mu\left(M_{1,1} + \left(\frac{1}{2}\right)^2 D_l + \left(\frac{1}{2}\right)^2 D_l\right) 2^{-2R_m} \\ &= \mu\left(M_{1,1} + \frac{1}{2}D_l\right) 2^{-2R_m} \\ &= \mu M_{1,1} 2^{-2R_m} + \mu^2 \frac{1}{2}\left(M_{1,3} + \frac{5}{8}D_K\right) 2^{-2(R_m+R_l)} \end{aligned}$$

- **Strategy 2:** Again according to the temporal estimation order in Figure 3.3, the distortion of the middle frame is:

$$D_m = \mu\sigma_m^2 2^{-2R_m} = \mu\left(M_{2,2} + \frac{1}{2}D_K\right) 2^{-2R_m}$$

  Then the distortion of each of the lateral frames reads:

$$\begin{aligned} D_l = \mu\sigma_l^2 2^{-2R_l} &= \mu\left(M_{1,1} + \frac{1}{4}D_m + \frac{1}{4}D_K\right) 2^{-2R_l} \\ &= \mu(M_{1,1} + \frac{1}{4}D_K) 2^{-2R_l} + \mu^2 \frac{1}{4}\left(M_{2,2} + \frac{1}{2}D_K\right) 2^{-2(R_m+R_l)} \end{aligned}$$

- **Strategy 3:** We start by estimating the distortion of the middle frame:

$$D_m = \mu\sigma_m^2 2^{-2R_m} = \mu\left(M_{2,2} + \frac{1}{2}D_K\right) 2^{-2R_m}$$

  Then, the distortion of the lateral frames is:

$$D_l = \mu\sigma_l^2 2^{-2R_l} = \mu\left(M_{1,3} + \frac{5}{4}D_K\right) 2^{-2R_l}$$

Then, it is possible to compute the total distortion of the WZFs for each strategy:

$$D_1 = D_m + 2D_l \tag{3.1}$$

In order to plot these three rate distortion functions, we have to estimate the quantities: σ_K^2, $M_{1,3}$, $M_{1,1}$ and $M_{2,2}$. We thus estimate these elements for each frame and then we calculate the average in order to have the general behavior of each sequence. We use 100 frames of the first camera for temporal coefficients and four times 8 frames at the same temporal instant for the view coefficients. Figure 3.4 presents these coefficients estimated on two multi-view test sequences, in the time direction and in the view direction.
Three remarks can be made:

- First, as expected, the motion/disparity prediction errors (M parameters), as well as the quantization errors, are much lower than the variance of the KFs (σ_K^2).
- Secondly, the estimation error is lower when the maximum distance (*i.e.*,the distance to the furthest frame) is small. Indeed, $M_{1,1} < M_{2,2} < M_{3,1}$.
- Finally, the estimation errors are more important for *breakdancer* sequence than for *ballet* sequence. We can thus expect worse results for this sequence and in general, estimating these prediction errors gives a good idea about the coding performances that may be expected for a given sequence.

The estimation of μ coefficients is based on a detailed rate distortion analysis presented in [Fraysse *et al.*, 2009]. We consider that the frames are coded at high bitrate and we assume that the KFs have a Gaussian distribution and the WZF errors have a Laplacian distribution. Note also that, in this reference are deduced rate-distortion models for theoretical sources and low bitrates. However, these are less practical to exploit, so here we keep with the classical high bitrate rate-distortion model. The μ coefficients can also be estimated from the real RD functions of the KFs or WZFs by performing a linear regression of the practical RD functions.

Using these estimated values, we plot the different RD functions for the two test sequences, *ballet* and *breakdancer*, in temporal and view directions. Figure 3.5 shows the experimental results and one can see that the best strategy is the second one.
We have thus the best solution for the one dimensional problem. The Figure 3.6 shows the proposed two dimensional solution corresponding to the previous analysis. Indeed, separately in the view direction and in the temporal direction, the best decoding strategy is the second one. The Figure 3.6 presents the decoding strategy, and the different estimations made for each WZF. For the first WZF to decode, we make the fusion between three estimations (temporal, inter-view and diagonal). For the second, we compute the fusion of temporal and view estimations.

3.1.3 Experimental validation

In this section we test the proposed approaches. We use again the two multi-view test sequences: *breakdancer* and *ballet*. In order to save some computation complexity, we reduce the spatial resolution to 256×192 after a low-pass filtering as it is done in [Areia *et al.*, 2007]. For both, the frame rate is 15 fps and we use the 8 cameras with the first 20 frames per view. The results are presented through rate-distortion performance. The

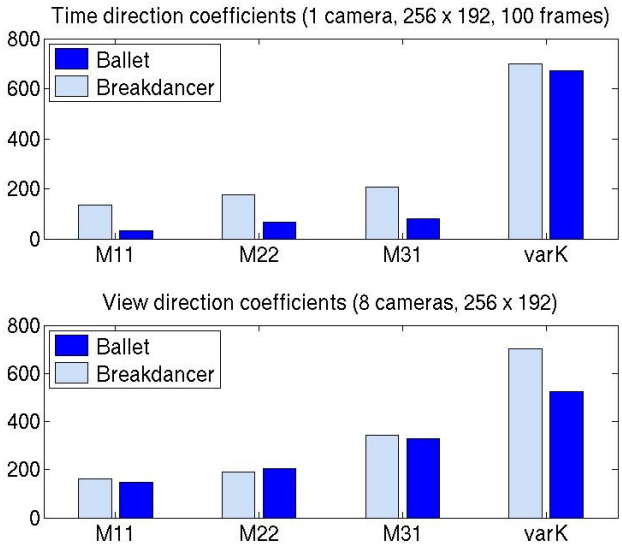


Figure 3.4: Values of the different dependency coefficients for "Ballet" and "Breakdancer" sequences.

Table 3.1: Complexity comparison (computation time in seconds per frame) at the encoder and at the decoder, for different schemes.

Scheme	Encoder	Decoder
H.264 Intra	0.25	0.03
Hyb2	0.21	5.11
Sym2	0.13	14.45
Sym4	0.07	17.46

rates presented are the total rates (WZF + KF) per camera (because the schemes used are symmetric) for the luminance component (as usual for WZ coding).

We present in Table 3.1, the computational complexity (time in seconds per frame), at the encoder and at the decoder for different schemes. This was measured on an "Intel Core 2 Duo" machine, 2.66 GHz, under Linux, for *breakdancer* sequence, on 5 views and 5 frames per view. The reported results are the average computation times per frame. The experimental results confirm that the Sym4 scheme is far less complex than Sym2 at the encoder (the encoding complexity of Sym4 represents only 50% of the Sym2 complexity and only 30% of the Intra configuration complexity), which is interesting for distributed video applications on low-power systems. The decoding complexity increasing is considered for the moment (here and in the literature) as a non-problem.

In experiments shown in Figure 3.7, we compare the Sym4 with the Sym2 and with the Hyb2 (see Figure 3.1 (c)). We notice that, when the performance of Sym2 is better than the Intra coding, the Sym4 is better than both Sym2 and Hyb2. This can be explained by

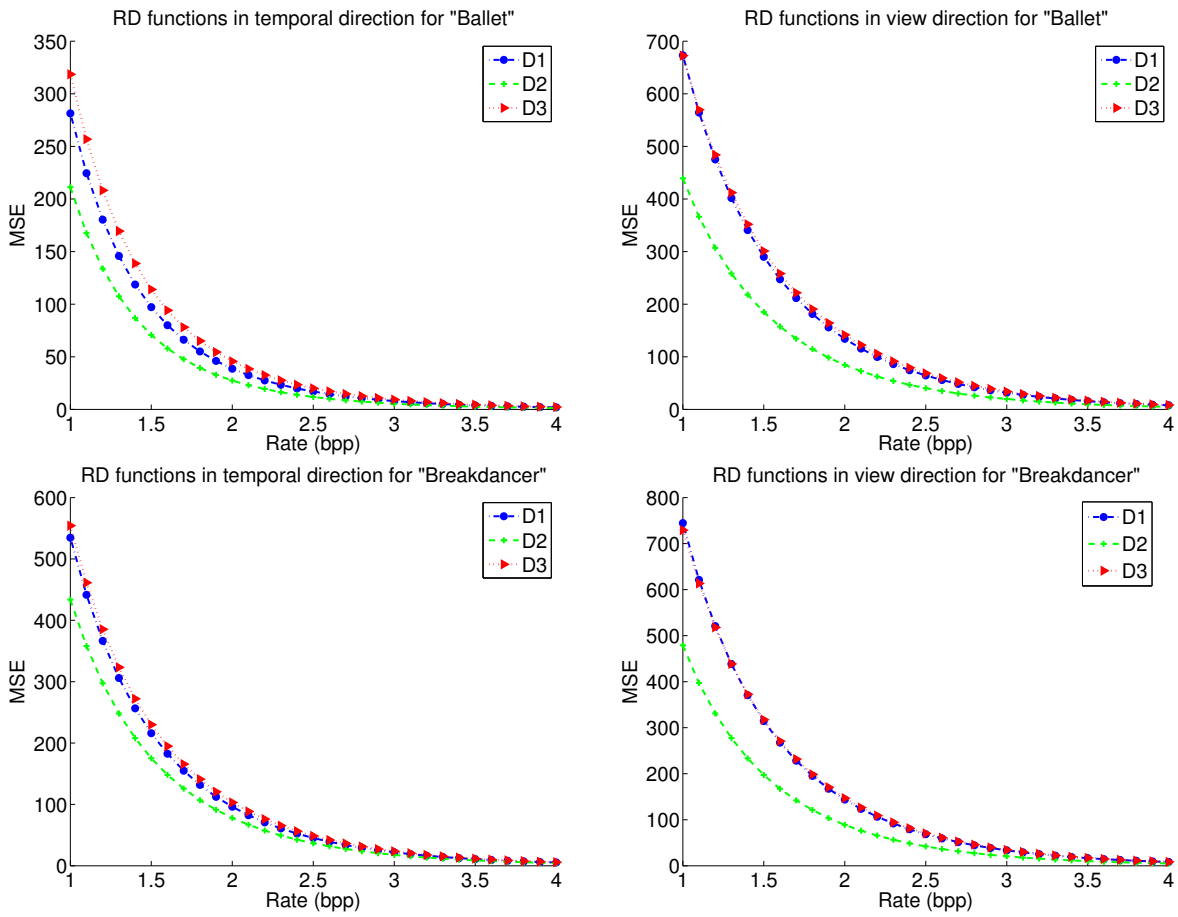


Figure 3.5: Rate-Distortion functions for the test sequences "Ballet" and "Breakdancer" (8 cameras, 256×192, 15 fps per view, average over 100 frames and 8 views). D_1, D_2 and D_3 are the distortions corresponding to the three estimation strategies.

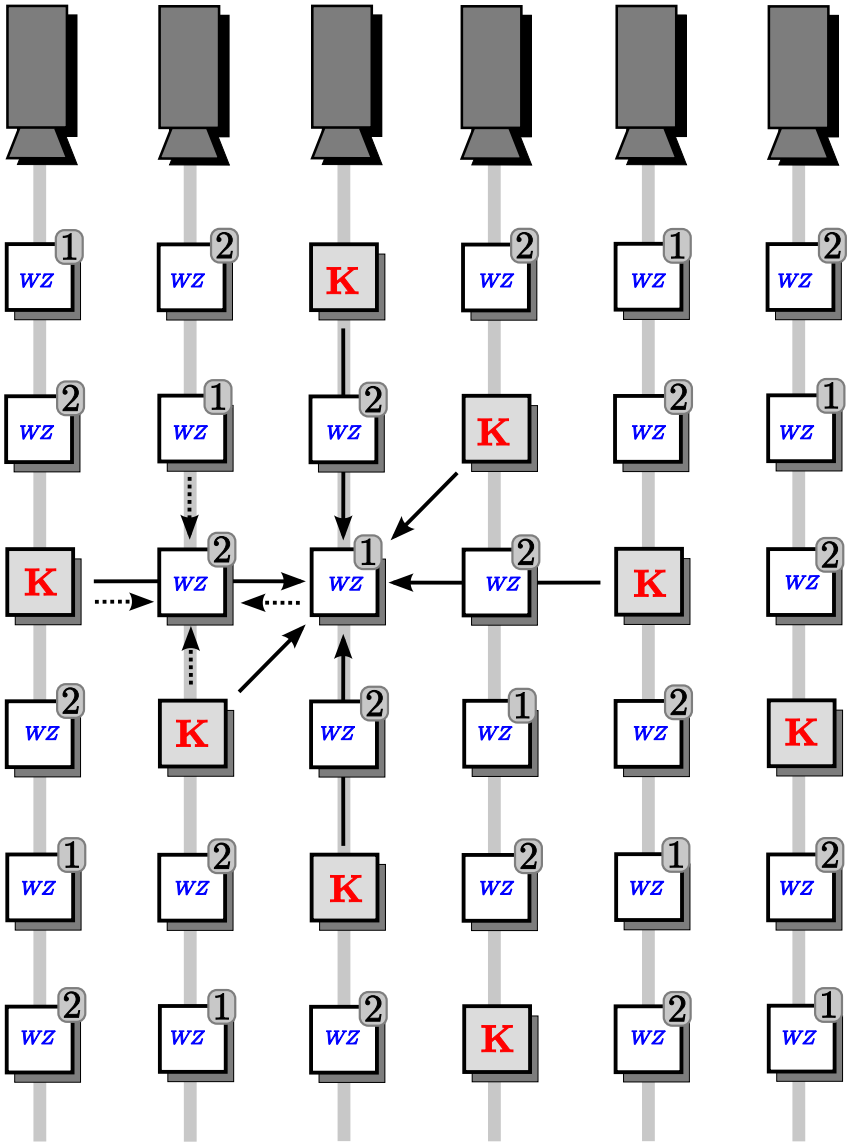


Figure 3.6: Decoding strategy for Sym4. The plain arrows represent the side information generation at the first step, and the dashed arrows at the second step.

the fact that the Intra frames are replaced by WZFs, using lower bit rates. However, for the *breakdancer* sequence, the coding efficiency is lower for the WZFs than for the Intra frames, and thus replacing KFs by WZFs degrades the performance. This explains why for this sequence Sym4 has lower performance than Hyb2, but we notice that Sym4 is better than Sym2. The results are interesting because they show the potential of using a scheme involving less KFs.

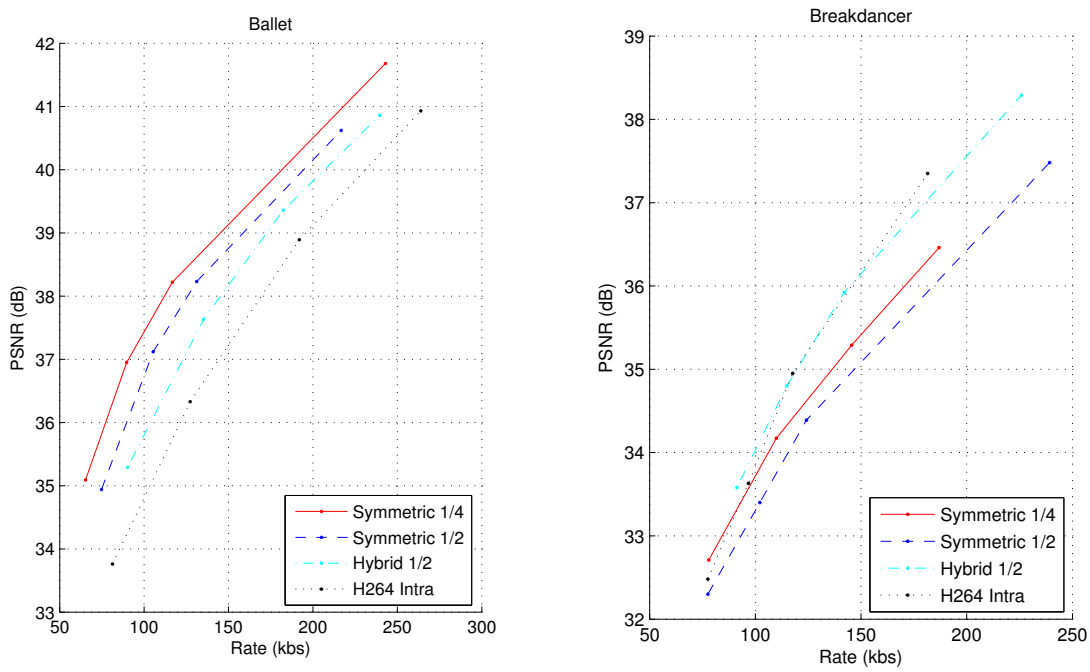


Figure 3.7: Comparison of the RD performance for *ballet* and *breakdancer* (8 cameras, 256×192, 15 fps per view) for different coding schemes.

3.2 Frame loss analysis

Once we have determine in the previous section an efficient decoding strategy for monoview DVC (Figure 3.3, strategy "2"), it would be interesting to study its behaviour in case of frame loss. This is what is performed in this section.

3.2.1 Context

Let us recall the adopted decoded strategy for a GOP size of 4. First the middle WZ frame, denoted by W_m, is decoded thanks to a side information generated using the two KFs, K_1 and K_2. Then, the lateral frames W_{l_1} and W_{l_2} are decoded using the reference frames and the decoded frame W_m. In Section 3.1.2, we proved that this decoding strategy is optimal between all possible decoding schemes. It has also been empirically used in [Aaron *et al.*, 2003]. We notice that the three kinds of frames play a different role in this

decoding process. [3]

3.2.2 Theoretical analysis

The expression of the average distortion in a GOP is: $D_T = \frac{1}{4}(D_K + D_{l_1} + D_m + D_{l_2})$. We recall that the general rate distortion function for a frame X can be approximated, at high bitrate, by

$$D_X = \mu\sigma_X^2 2^{-2R_X}, \tag{3.2}$$

where R_X is the allocated rate in bits per pixel, σ_X^2 the original variance of the frame X, and μ a constant depending on the source distribution (see Section 2.4.1). In this section we study the expression of the GOP distortion for several case of figure: no loss, loss of a key frame, loss of a middle WZ frame and finally, loss of a lateral WZ frame.
Case of a lossless transmission: This case has already been studied in Section 3.1.2. We do not give the detail of the calculation, we thus only briefly recall the obtained distortions:

$$D_K = \mu_K\sigma_K^2 2^{-2R_K} \tag{3.3}$$

$$D_m = \mu_m\left(M_{2,2} + \frac{1}{2}D_K\right)2^{-2R_m} \tag{3.4}$$

$$D_{l_i} = \mu_l\left(M_{1,1} + \frac{1}{4}D_K + \frac{1}{4}D_m\right)2^{-2R_{l_i}}. \tag{3.5}$$

We obtain the average distortion of a GOP using:

$$D_T = \frac{1}{4}(D_K + D_m + 2D_{l_i}). \tag{3.6}$$

Loss of parity bits for W_{l_1}: if the parity bits used to decode the frame W_{l_1} are lost, the estimation error can not be corrected. Thus, we have $R_{l_1} = 0$. The distortion of the frame W_{l_1} is that of its corresponding SI and can be expressed as:

$$D_{l_1}^* = \mu_l\left(M_{1,1} + \frac{1}{4}D_K + \frac{1}{4}D_m\right). \tag{3.7}$$

The distortion of the KF, as well as W_m and W_{l_2}, remain unchanged and are expressed as in Equantions (3.3), (3.4) and (3.5).
The average GOP distortion becomes:

$$D_T^l = \frac{1}{4}(D_K + D_m + D_{l_1}^* + D_{l_2}). \tag{3.8}$$

Loss of parity bits for WZ_m: in this case, the distortion of the KF is as in (3.3), and the distortion of the W_m frame is:

$$D_m^* = \mu_m\left(M_{2,2} + \frac{1}{2}D_K\right), \text{ since } \quad R_m = 0. \tag{3.9}$$

[3]The material in this section was published in:

- T. Maugey, T. André, B. Pesquet-Popescu, and J. Farah, "Analysis of error propagation due to frame losses in a distributed video coding system," in *Proc. Eur. Sig. and Image Proc. Conference (EUSIPCO)*, Lausanne, Switzerland, Aug. 2008.

Therefore, the distortion of the W_{l_i} frames, for $i \in \{1, 2\}$, becomes:

$$D^*_{l_i} = \mu_l \left(M_{1,1} + \frac{1}{4} D_K + \frac{1}{4} D^*_m \right) 2^{-2R_{l_i}}. \tag{3.10}$$

We have the following average distortion of a GOP:

$$D^m_T = \frac{1}{4}(D_K + D^*_m + 2D^*_{l_i}). \tag{3.11}$$

Loss of a Key Frame: When K_1 (or K_2) is lost, before decoding the corresponding GOP, this frame needs to be estimated using other KFs supposed to be well received (the two located at a distance of 4 frames before and after the current lost KF). The corresponding estimation error variance is $\sigma^2_{e_K}$. Therefore, the distortion of the KF is:

$$D^*_K = \mu^*_K \sigma^2_{e_K} 2^{-2R_K} \quad \text{, with} \quad R_K = 0$$

$$D^*_K = \mu^*_K \left(M_{4,4} + \frac{1}{2} D_K \right). \tag{3.12}$$

Thus, the distortion of the W_m frame will be:

$$D^*_m = \mu_m \left(M_{2,2} + \frac{1}{4} D^*_K + \frac{1}{4} D_K \right) 2^{-2R_m}. \tag{3.13}$$

and the distortion of the W_{l_1} and W_{l_2} frames modifies accordingly:

$$D^*_{l_1} = \mu_l \left(M_{1,1} + \frac{1}{4} D^*_K + \frac{1}{4} D_m \right) 2^{-2R_{l_1}} \tag{3.14}$$

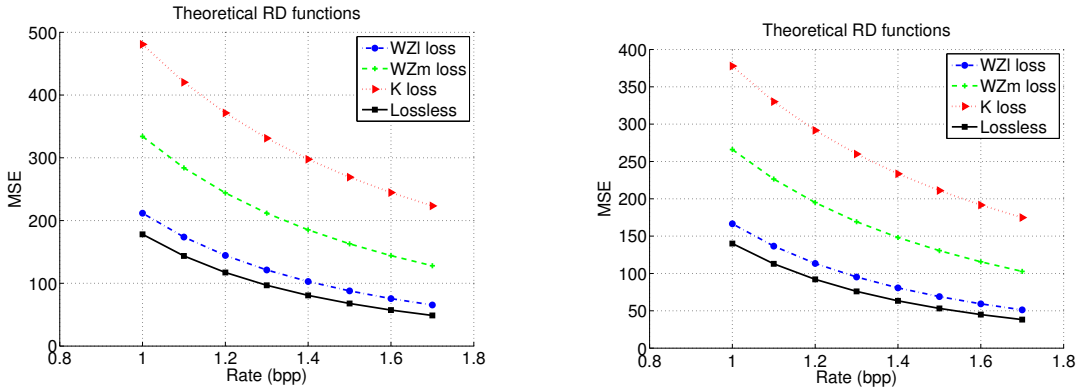
$$D^*_{l_2} = \mu_l \left(M_{1,1} + \frac{1}{4} D^*_m + \frac{1}{4} D_K \right) 2^{-2R_{l_2}}. \tag{3.15}$$

We have the following average GOP distortion in this case:

$$D^K_T = \frac{1}{4}(D^*_K + D^*_m + D^*_{l_1} + D^*_{l_2}). \tag{3.16}$$

The motion interpolation errors ($M_{1,1}$, $M_{2,2}$, $M_{4,4}$) are experimentally estimated. These errors, as well as σ^2_K, have been estimated with the test sequences *foreman* (QCIF, 30 fps, 200 frames) and *coastguard* (QCIF, 30 fps, 150 frames). The estimation of μ coefficients was firstly based on a detailed rate distortion analysis presented in [Fraysse *et al.*, 2009], as in Section 3.1.2, but were finally experimentally determined using a linear regression of practical RD functions. Moreover, we experimentally established that the rates for the four frames must be different in order to have a uniform decoding quality in a GOP: if we consider a rate R in bpp for the KF, the rate for the W_m frame is arbitrary taken $R/2$ and for the W_l as $R/4$. These ratios were adopted for the theoretical plots (Figure 3.8) where we present the average rate in bpp.

Because of several approximations assuming high bitrate hypotheses (detailed in the previous chapter), the values of the theoretical rate distortion function are bigger than expected for low bitrate and we only present the curves at high bitrate (above 1 bpp). However, these plots still allow more interesting remarks. In Figure 3.8, we notice the importance of the error propagation phenomenon. Indeed, for both video sequences, the loss of a KF propagates over the entire GOP and leads to a much higher distortion than in the case of a W_m loss, which in turn induces a more important distortion than that caused by a W_l frame loss. These theoretical results thus illustrate the fact that an error occurred in a K or W_m frame will spread over the other frames when using that biased frame as a reference frame.



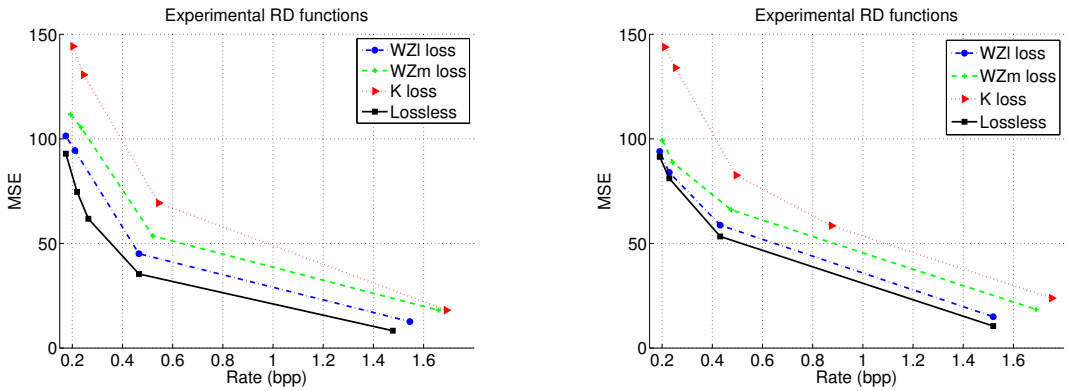
(a) *foreman*, QCIF, 30 fps, 200 frames. (b) *coastguard*, QCIF, 30 fps, 150 frames.

Figure 3.8: Theoretical rate-distortion functions, corresponding to the lossless case and to the three loss situations, for (a)*foreman* and (b)*coastguard* sequences.

3.2.3 Experimental validation

In this section, we compare the experimental and theoretical rate-distortion functions in the same frame loss conditions as those considered in the theoretical study (Figure 3.8) in previous section. Practical WZ coding was obtained with a DISCOVER scheme. Experiments were run on the same test video sequences, *foreman* and *coastguard*. The results presented in Figure 3.9 correspond, at each bitrate, to the average distortion of the entire sequence. For each loss type, every GOP in the sequence is affected by the loss (e.g., for a W_m or W_l loss, one over four frames in the sequence are lost). If the lost frame is a WZF, its parity bits are transmitted but cannot be exploited by the decoder. For the WZ frames losses, no concealment is performed at the decoder. But if the lost image is a KF, the frame is estimated at the decoder using the two closest KF.
Two main remarks can be done regarding these experimental plots. First, we are able to see in the obtained curves the error propagation caused by a frame loss. Indeed, the experiments show that if a frame is used to generate the side information for other WZFs, its loss will deeply affect the decoding performances. The second remark concerns the similarity between the theoretical and experimental plots. Indeed, the theoretical plots have predicted the relative importance of the frame losses (K,W_m, W_l) at high bitrate. One can see in the experiments that this prediction is also true at low bitrate. The proposed theoretical approach can thus be used in similar situations in order to improve the decoding performances.

Moreover, we present another experimental result which analyzes the evolution of the decoder behavior through time and compares the case of lossless transmission to the case where the transmission is randomly affected by frame losses (Figure 3.10). In such a decoding scheme, it is interesting to study the side information evolution linked with the rate per frame evolution. Indeed, the final PSNR of each frame is almost equal for a lossy or a lossless transmission, since the rate for a WZF will increase in order to correct the errors using the parity bits. Then, if the estimation error is bigger, the requested parity bits will be more numerous, but the decoded frame will have almost the same PSNR. In Figure 3.10 (up), we present the evolution of the side information quality (for the KFs, we represent the PSNR of the decoded frame). In Figure 3.10 (bottom), the evolution of the transmitted rate per frame is presented. The experiments were run on the *coastguard*



(a) *foreman*, QCIF, 30 fps, 200 frames. (b) *coastguard*, QCIF, 30 fps, 150 frames.

Figure 3.9: Experimental rate-distortion functions, corresponding to the lossless case and to the three loss situations, for (a)*foreman* and (b)*coastguard* sequences.

(QCIF, 30 fps) sequence with the first 97 frames. For the lossy transmission (plain plots), the frame losses occurred randomly. The vertical lines represent the moments when the losses occurred (solid lines for K losses, dashed lines for W_m losses, and dotted lines for W_l losses). One can notice that the rates for KFs and WZFs do not exactly correspond to the ratios indicated in the previous section. Indeed, they have been established experimentally taking into account a larger number of frames.
The obtained curves confirm the previous remarks on the relative importance of the frame losses (K, W_m, W_m). Indeed, we can see that a K loss affects the 6 other frames around it, i.e. their SI PSNR is lower and their rate per frame is bigger. Besides, the loss in SI PSNR and the increase in the requested data rate are larger for the closest neighbors than the rest of the GOP. This proves that the error propagation influence due to frame loss decays with time (in both directions). On the other side, a W_m loss affects only two frames around it, whereas a W_l loss does not affect any other frame. In fact, a W_l loss is not visible on the presented curves because only the reconstruction is affected in this case and it does not concern the transmission rate or the SI PSNR.

3.3 Backward channel suppression

3.3.1 Introduction

3.3.1.1 Motivations and related problems of rate control at the encoder

We previously mentioned that the main problem of actual DVC schemes is the presence of a feedback loop, thus forcing a real time decoding and negligible transmission times, not conceivable in practice. This backward channel is employed to create a communication between the turbo encoder and the turbo decoder. More precisely, after the reception of a first stream of parity bits (the parity bits are divided into a certain number of chunks), the turbo decoder performs the corresponding bitplane decoding. Then it estimates the error probability for it, and if this one is greater than a threshold (arbitrary fixed here at 10^{-3} [Brites *et al.*, 2008]), the turbo decoder requests another parity bits stream, *via* the backward channel. This operation is repeated until the error probability becomes lower

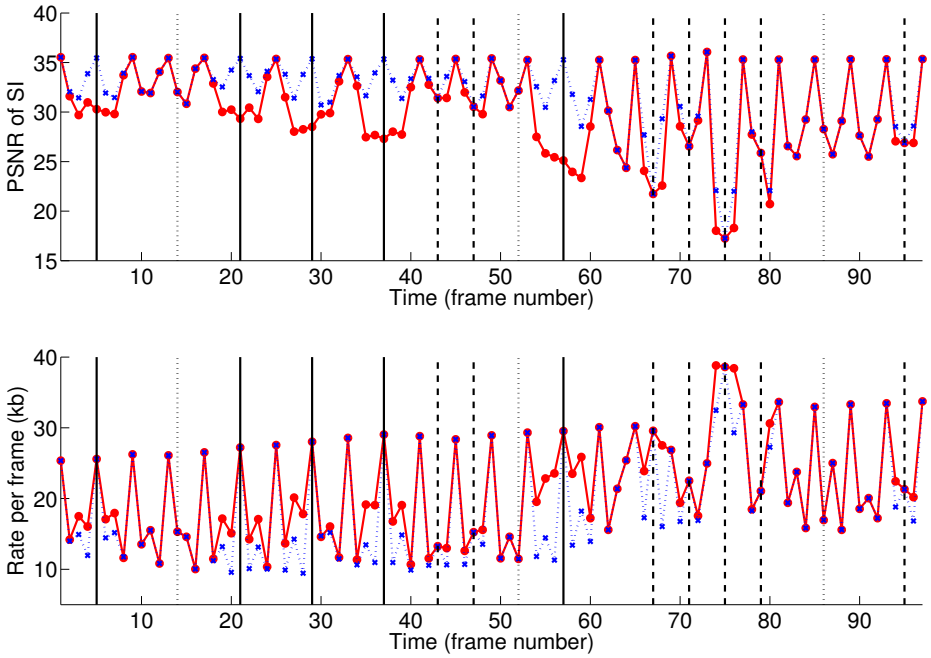


Figure 3.10: Evolution of the side information PSNR (up) and of the rate per frame (bottom) through time. The dotted curves correspond to a lossless transmission and the plain curves correspond to the case where the transmission is randomly affected by frame losses. The KF losses (resp. W_m and W_l) are represented by vertical plain lines (resp. dashed and dotted lines).

than the threshold, or if a maximum number of parity bits requests[4] has been reached. Knowing the decoding mechanism which is performed for each bitplane of each band, the use of the backward channel is obvious: allowing a transmission with an optimal rate, *i.e.*, the minimum rate required for a reconstruction with a bit error probability under 10^{-3}. Then, the suppression of this backward channel can degradate the rate-distortion performance.

3.3.1.2 Existing rate estimation algorithms

There exists not so many solutions for rate control at the encoder and some of them are developed for a specific context, quite different from our transform domain scheme. For this reason we just mention here the methods developed by Morbee *et al.* [Morbee *et al.*, 2007] and by Yaacoub *et al.* [Yaacoub *et al.*, 2008], using pixel-domain DVC scheme.

In our context, *i.e.*,a scheme inspired by Discover, working in the DCT domain, and describing the WZ information with bitplanes, only three methods were proposed. All of them calculate for each bitplane the number of parity information to send and use a GOP size of 2 in their test.

Brites and Pereira's algorithm [Brites, Pereira, 2007] estimates the bitplane entropy by considering the error probability based on the Laplacian error distribution modelled with a coarse version of the side information (for example the average of the reference frames or a fast motion interpolation). The algorithm deduces from this entropy a quantity of information allocated to the current bitplane. Because a rate underestimation could have dramatical consequences on the final performances, Brites and Pereira proposed to add a term which takes into account the error propagation along the bitplanes. Whereas significant losses are conceivable, this method presents a too high dependency to the coarse side information calculated at the encoder. Indeed, the gap between a simple key frame average and a fast motion interpolation is high (except for *hall monitor* sequence which has almost no motion). Moreover, the performance quality seems to also strongly depend on the additional term, and its calculation is not precisely explained in the paper. It is thus difficult to determine if this additional term needs to estimate some parameters or not.

Sheng *et al.*. [Sheng *et al.*, 2008; Sheng *et al.*, 2010] have proposed a very similar approach, where the number of parity bits needed at the decoder is estimated based on the correlation noise estimation (*i.e.*, the Laplacian distribution parameter used to model the side information error).

More recently, Halloush and Radha [Halloush, Radha, 2010] proposed a quite different approach. They estimate bitplane by bitplane the parity rate based on the Hamming distance between the previous key frame and the current WZ frame. They obtained losses of equivalent order of magnitude.

Moreover Kubasov *et al.* [Kubasov *et al.*, 2007a] also make a rate estimation at the encoder. However, their purpose is no longer to avoid the backward channel but to reduce the decoding complexity by sending an estimated rate for each bitplane and by completing it by requesting the missing parity bits with the return loop. They estimate the rate, as

[4] In fact, in some implementations, another criterion for the bitplane decoding stop is when the decoding does not converge, *i.e.*, when the error probability does not decrease after a certain number of requests.

Brites and Pereira, by integrating the Laplacian distribution over the bins, and using this value to calculate the bitplane conditional entropy. If the methods which aim at getting rid of the return loop must not perform a rate underestimation, the rate estimation technique of Kubasov *et al.* aim at having no overestimation. Consequently, even if the techniques are similar, the target are quite different.

3.3.1.3 Hypotheses and main idea of the proposed approach

All of the existing methods have done the choice to directly estimate the parity rate bitplane by bitplane without firstly estimating a global frame rate. In our opinion, it would be more precise to consider that the problems related to backward channel suppression are twofold. Firstly, the encoder needs to estimate the total rate per frame (the sum of the parity bits required for all the bitplanes of all bands), and secondly, the encoder has to estimate the distribution of this total rate among all the bitplanes of all the bands.
In this Section 3.3, we present a solution to this problem. More precisely, we present in Section 3.3.2 how we estimate the rate per frame, based on the previously introduced model. Then, in Section 3.3.3, we present our approach to estimate the number of parity bits to send for each bitplane of each band.

While the existing rate control algorithms are only tested with a GOP size of 2, we think that it would be more challenging if the proposed technique was tested for a configuration where the ratio of WZ frames is larger than 1/2. More precisely, we adopt a structure with a GOP length equal to 4: one reference frame followed by three WZ frames, where the different WZ frames do not play the same role inside the GOP. The optimal decoding order was proved in Section 3.1.2 and is presented in Figure 3.3 (b).

In the following, we keep the same notations as above for K, W_m, W_l, D_K and D_m, D_l, R_K, R_m and R_l (Section 3.2.1).

3.3.2 Frame rate estimation

The first problem of backward channel suppression is to predict at the encoder the total rate for each frame of the sequence. We propose to calculate for each frame the rate needed to obtain an homogeneous decoded frame distortion along the GOP (and then along the sequence). First, in Section 3.3.2.1 we introduce, based on the model of Chapter 2, an expression of the distortion for each frame of a GOP. Then, in Section 3.3.2.2, we deduce an expression of the theoretical rates (R_K, R_m and R_l). Then (in Section 3.3.2.3) we explain how to estimate the allocated rates based on the theoretical formulas. Finally, in Section 3.3.2.4, we compare the predicted rate with the experimental rate (DISCOVER with a return loop).

3.3.2.1 Rate expression

Using Equation (2.8), the distortion of each frame of the GOP can be determined. We recall here the expressions of the distortion:

$$\begin{aligned} D_K &= \mu_K \sigma_K^2 2^{-2R_K} \\ D_m &= \mu_m \left(M_{2,2} + \frac{1}{2} D_K \right) 2^{-2R_m} \\ D_l &= \mu_l \left(M_{1,1} + \frac{1}{4} D_K + \frac{1}{4} D_m \right) 2^{-2R_l}. \end{aligned}$$

3.3.2.2 Homogeneous distortion inside the GOP

Several criteria can be adopted for determining the optimal rate-distortion tradeoff. In the proposed approach, we choose a simple and justified (corresponding to a constraint for good visual quality) criterion: the distortion along the sequence must be constant. We can thus add the following constraint on the previous equations:

$$D_K = D_m = D_l$$

in order to have the same distortion along the GOP. Let us formulate the WZ rates as a function of the key frame rate, R_K. First, the middle WZ frame rate, R_m is obtained by writing

$$\begin{aligned} D_m &= D_K \\ \mu_m \left(M_{2,2} + \frac{1}{2} D_K \right) 2^{-2R_m} &= D_K \\ R_m &= \frac{1}{2} \log_2 \left(\frac{\mu_m \left(M_{2,2} + \frac{1}{2} D_K \right)}{D_K} \right). \end{aligned}$$

With the same approach, we obtain the lateral WZ frame rate

$$\begin{aligned} D_l &= D_K \\ \mu_l \left(M_{1,1} + \frac{1}{4} D_K + \frac{1}{4} D_m \right) 2^{-2R_l} &= D_K \\ R_l &= \frac{1}{2} \log_2 \left(\frac{\mu_l \left(M_{1,1} + \frac{1}{2} D_K \right)}{D_K} \right). \end{aligned} \tag{3.17}$$

Finally, we obtain two rate expressions which are directly determined by the key frame distortion. In other words, after the choice of the key frames quality (*i.e.*, after adjusting the QP), the rates of the WZ frames are directly determined.

3.3.2.3 Practical approach

At this step, we have the explicit expressions of the rates for each frame inside the GOP. However, these expressions still contain several parameters which need to be estimated.

- The μ coefficients depend on the source distributions, they can be theoretically determined as explained in [Fraysse *et al.*, 2009]. In our practical framework, we experimentally obtain (by linear regression) the μ parameters based on experimental rate distortion performances.

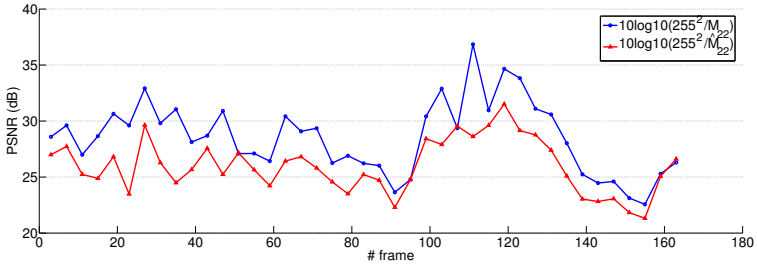
- The $M_{1,1}$ and $M_{2,2}$ coefficients correspond to the interpolation errors in case of zero-distortion reference frames. By definition, they cannot be calculated at the encoder because of the DVC principle and because of the complexity of motion interpolation methods. For this problem, we consider that the two reference frames are available at the encoder and we perform a simple average (low computational complexity) between them. The use of the key frames at the encoder may be arguable, as it opposes the distributed source coding main framework. However, it is a classical liberty taken in the literature [Ascenso, Pereira, 2007], [Morbee *et al.*, 2007], [Sheng *et al.*, 2010], [Halloush, Radha, 2010] and as long as it remains non complex, it is acceptable for practical applications. Moreover, in the hypotheses of DSC, the encoder needs to know the exact correlation between the two sources. In our case, the correlation information is mainly given by these M coefficients. Obviously, the true M values cannot be available in practice at the encoder, but they can be estimated. That is why we estimate the M coefficients by $\hat{M}$, the distortion of the average between the two reference frames. In Figure 3.11, one can observe the evolutions of the true PSNR associated to M and of the estimated PSNR associated to $\hat{M}$ for *foreman* and *soccer* sequences. It can be highlighted that the estimated PSNR evolution is quite similar to real PSNR one, which is promising for rate estimation.

- The variance σ_K^2 can be directly estimated at the encoder (this information is easily available). Logically, we should not consider that this information would be available at the WZ encoder, because of the distributed source coding spirit. However the liberty of accessing to the key frames informations has already been taken and justified in the previous point, therefore, we consider σ_K^2 information available. In fact, the results do not change very much wether the variance is constant or not.

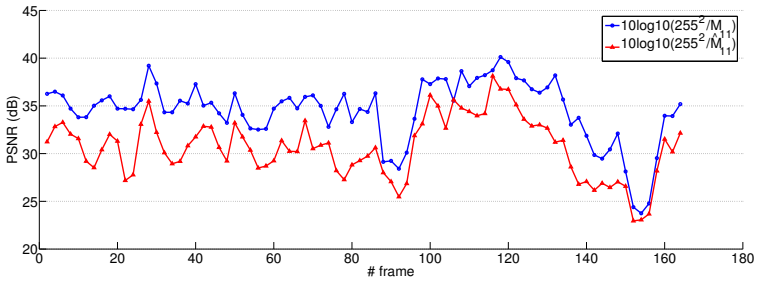
3.3.2.4 Experiments

For several sequences, we compare the predicted rate to the experimental rate obtained with the DISCOVER scheme with a return loop. In the first column of Figure 3.12 (respectively second column of Figure 3.12), the plots correspond to the normalized rates (for a better readability, the rates have been divided by their maximum) for the middle WZ frames W_m (respectively the lateral WZ frames W_l). Note that the maximum value for the theoretical and the experimental rates are not the same. This comes from the approximation of the proposed model. These multiplying coefficients need to be offline estimated and vary from a sequence to another.

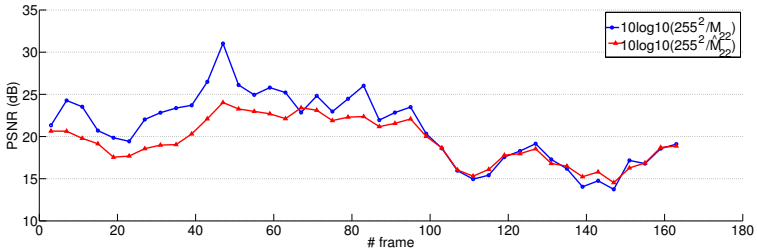
It can be seen that the predicted rate corresponds to the experimental rate. Even if there is still a small imprecision, the high variations are well estimated. To confirm this observation, we have calculated the percentage of underestimated and overestimated frame rates (see Table 3.2). Firstly, one can remark that the rates are mainly overestimated, which is justified by the fact that underestimating the number of parity bits to send sensibly damage the reconstruction. Furthermore, one can observe that the results are quite acceptable, because a very few percentage of frame have a $|\Delta\text{Rate}|>10\%$. In [Sheng *et al.*, 2008], between 9 and 15% of the frames have a $|\Delta\text{Rate}|>30$kbs for QCIF sequences. In our tests, where a difference of 10% corresponds approximately to a error of 20kbs, one can see that never more than 3% of the frames have a $|\Delta\text{Rate}|>20$kbs, which is sensibly more acceptable. This is the advantage of having a global frame vision when allocating



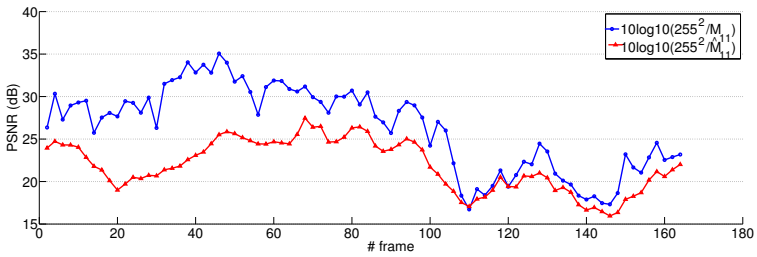
(a) W_m of *foreman* sequence



(b) W_l of *foreman* sequence

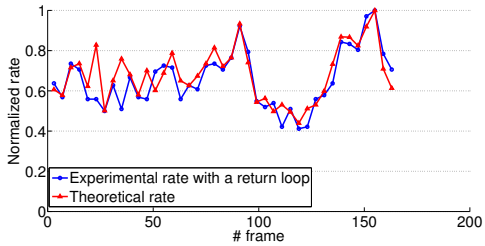


(c) W_m of *soccer* sequence

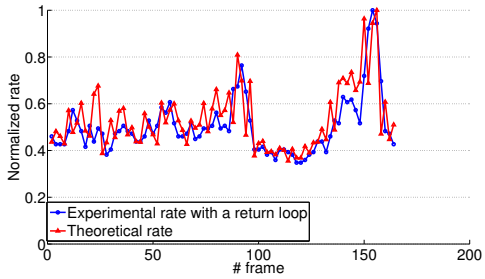


(d) W_l of *soccer* sequence

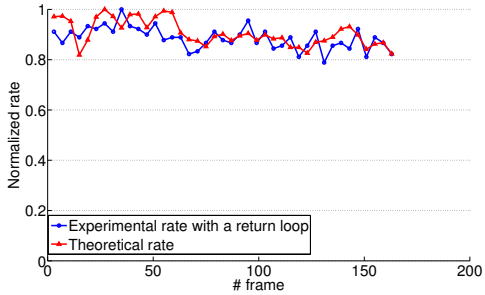
Figure 3.11: Comparison between the true PSNR associated to M and the estimated PSNR associated to $\hat{M}$ for two CIF sequences (352×288, 30 frame per second).



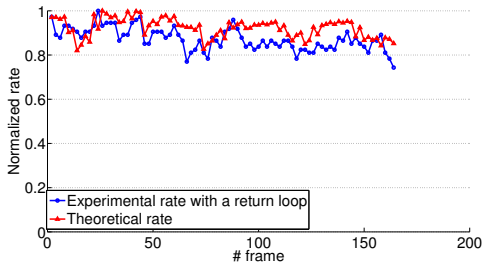
(a) W_m of *foreman* sequence (CIF)



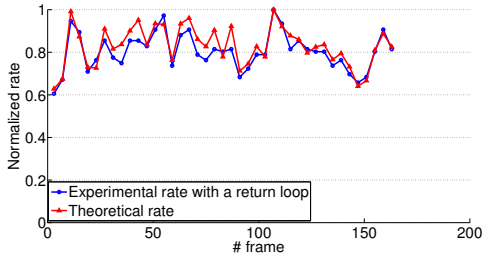
(b) W_l of *foreman* sequence (CIF)



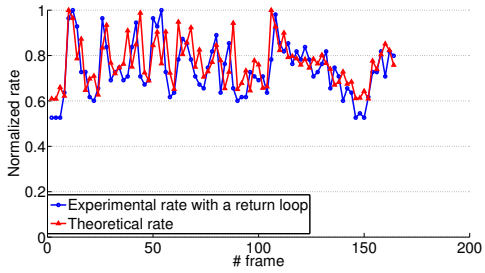
(g) W_m of *city* sequence (CIF)



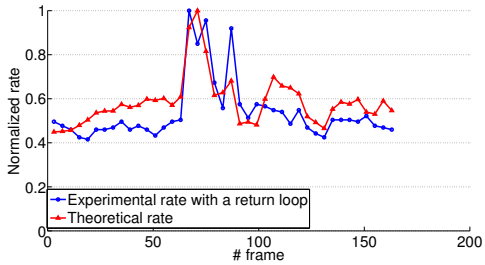
(h) W_l of *city* sequence (CIF)



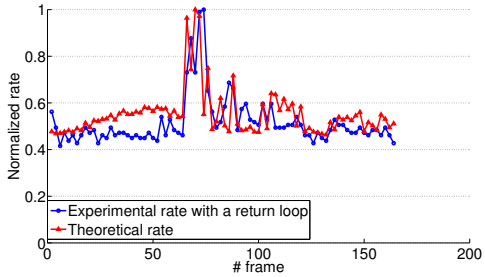
(g) W_m of *silent* sequence (QCIF)



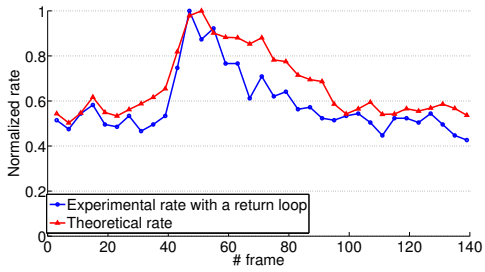
(h) W_l of *silent* sequence (QCIF)



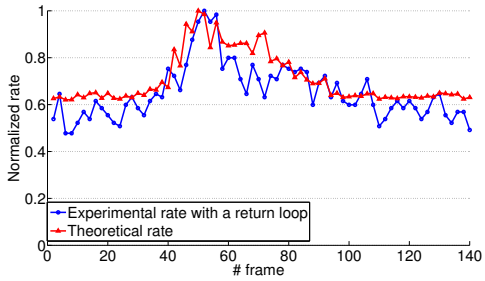
(g) W_m of *coastguard* sequence (QCIF)



(h) W_l of *coastguard* sequence (QCIF)



(g) W_m of *suzie* sequence (QCIF)



(h) W_l of *suzie* sequence (QCIF)

Figure 3.12: Comparison between the normalized experimental and theoretical rates.

	← underestimation		good estimation			overestimation →	
ΔRate (%)	(-∞ -10)	[-10 -5)	[-5 -2)	[-2 2]	(2 5]	(5 10]	(10 +∞)
foreman (CIF)	1.6	8.9	21.9	44.7	11.3	9.7	1.6
city (CIF)	0	0	3.2	55.2	40.6	0.8	0
silent (QCIF)	0	0.8	10.5	53.6	21.1	12.1	1.6
coastguard (QCIF)	0.8	2.4	12.1	23.5	30.0	28.4	2.4
suzie (QCIF)	0	0	2.8	37.1	29.5	27.6	2.8
Average in %	2.9		79.7			17.4	

Table 3.2: Percentage of frames of a sequence whose ΔRate (difference between theoretical and experimental rate in %) is included in the range.

the rate. The next step is to share this rate among the bitplanes. This is the goal of the method presented in next section.

3.3.3 Bitplane rate estimation

Knowing the total bitrate needed for a WZ frame, the next step is to determine the number of parity bits which have to be sent band by band, and bitplane by bitplane in order to allow a correct turbo decoding. Let us first recall the WZ frame encoding process (Section 3.3.3.1), before presenting the ideas of the proposed approach (Section 3.3.3.2) and finally testing it (Section 3.3.3.3).

3.3.3.1 Wyner-Ziv frame encoding

While the frame rate estimation (proposed in Section 3.3.2) does not completely depend on the precise implementation of the adopted coder (for example LDPC codes can replace turbocodes, etc.), the bitplane rate estimation is directly correlated to the chosen WZ encoding technique. That is why we quickly recall in this subsection the WZ encoding process, described in [Artigas *et al.*, 2007a].
At the encoder the WZ frames are 4×4 DCT transformed, decomposing the frame into 16 frequency bands. Then, the coefficients of each band are quantized. Knowing that low frequency coefficients have a larger dynamics than the high frequency ones, the quantization steps must depend on the band. For each band, a certain number of levels, 2^M, is fixed, obtaining then a number of M bitplanes associated to this band (and a corresponding quantization step). In [Brites *et al.*, 2006b], Brites *et al.* present the Discover quantization approach. They use 8 quantizers, represented by their QI, (QI=1 corresponds to the lowest bitrate, and QI=8 to the highest bitrate), and for each of them, they fix the number of bitplanes for each band. In other words, for each QI we have several rates $r_{b,bp}$ to estimate, as represented in Table 3.3 for QI = 8 (where b index corresponds to the band index, and bp denotes the bitplane level).

3.3.3.2 Proposed algorithm

As explained in the previous section, the problem of bitplane rate estimation consists in determining how to share the total frame bitrate, $\hat{R}$ (estimated on the basis of the proposed rate-distortion model), between the bitplane rates $r_{b,bp}$. In other words, the purpose is to choose the rates $r_{b,bp}$ under the constraint $\sum_{b,bp} r_{b,bp} = \hat{R}$.

	bitplane						
band	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$	$r_{1,4}$	$r_{1,5}$	$r_{1,6}$	$r_{1,7}$
	$r_{2,1}$	$r_{2,2}$	$r_{2,3}$	$r_{2,4}$	$r_{2,5}$	$r_{2,6}$	0
	$r_{3,1}$	$r_{3,2}$	$r_{3,3}$	$r_{3,4}$	$r_{3,5}$	$r_{3,6}$	0
	$r_{4,1}$	$r_{4,2}$	$r_{4,3}$	$r_{4,4}$	$r_{4,5}$	0	0
	$r_{5,1}$	$r_{5,2}$	$r_{5,3}$	$r_{5,4}$	$r_{5,5}$	0	0
	$r_{6,1}$	$r_{6,2}$	$r_{6,3}$	$r_{6,4}$	$r_{6,5}$	0	0
	$r_{7,1}$	$r_{7,2}$	$r_{7,3}$	$r_{7,4}$	0	0	0
	$r_{8,1}$	$r_{8,2}$	$r_{8,3}$	$r_{8,4}$	0	0	0
	$r_{9,1}$	$r_{9,2}$	$r_{9,3}$	$r_{9,4}$	0	0	0
	$r_{10,1}$	$r_{10,2}$	$r_{10,3}$	$r_{10,4}$	0	0	0
	$r_{11,1}$	$r_{11,2}$	$r_{11,3}$	0	0	0	0
	$r_{12,1}$	$r_{12,2}$	$r_{12,3}$	0	0	0	0
	$r_{13,1}$	$r_{13,2}$	$r_{13,3}$	0	0	0	0
	$r_{14,1}$	$r_{14,2}$	0	0	0	0	0
	$r_{15,1}$	$r_{15,2}$	0	0	0	0	0

Table 3.3: Rate matrix per band and per bitplane for QI= 8

The proposed algorithm can be summed up as:

1. The encoder performs a coarse estimation of the side information at the decoder (in practice the average of the reference frames computed previously for the choice of the total bitrate is used).

2. Band by band, and bitplane by bitplane, the encoder calculates the Hamming distance, $d_{b,bp}^{Ham}$ (number of different bits, for two vectors x_i and y_i, $i \in [1, N]$ $d^{Ham} = \sum_{i=1}^{N} x_i \oplus y_i$, where $\oplus$ is the logical XOR), between the bitplane of the original frame and the corresponding bitplane in the average estimation.

3. Deducing from the Hamming distances computed previously, the percentage, $p_{b,bp}^{\%}$, of the total rate to be affected, band by band, and bitplane by bitplane by the formula:

$$p_{b,bp}^{\%} = \frac{d_{b,bp}^{Ham}}{\sum_{b,bp} d_{b,bp}^{Ham}}.$$

4. The encoder computes the rates:

$$r_{b,bp} = p_{b,bp}^{\%} \cdot \hat{R}.$$

5. The encoder then adds a security rate on the more significant bitplanes. This security is a multiplying factor which is high for the most significant bitplanes, and which regularly decreases until the last bitplane. It depends on the QI adopted for the WZ frame. In our experimental results we set the exact values of this multiplying coefficient offline for each video, which obviously cannot be done in practice.

Step 5 was added because the first experiments have shown that even if the bitplane rates are in general well estimated, a small underestimation of a rate at this level could

		Avg rate/frame (kb)			Avg PSNR (dB)		
		Discover	Prop.	Δ	Discover	Prop.	Δ
(CIF)	*foreman*	12.31	15.96	3.65	30.51	30.12	−0.39
	city	18.31	26.59	8.28	26.83	26.49	−0.34
(QCIF)	*silent*	3.81	4.64	0.83	29.27	29.11	−0.16
	coastguard	4.18	5.13	0.95	27.80	27.95	−0.15
	suzie	3.23	4.33	1.10	32.49	32.16	−0.33

Table 3.4: Average (Avg) rate/frame (kb) and PSNR (dB) comparison between Discover and proposed no feedback scheme (denoted by Prop. above) performances, for several sequences, when the key frames are quantized with a QP of 40.

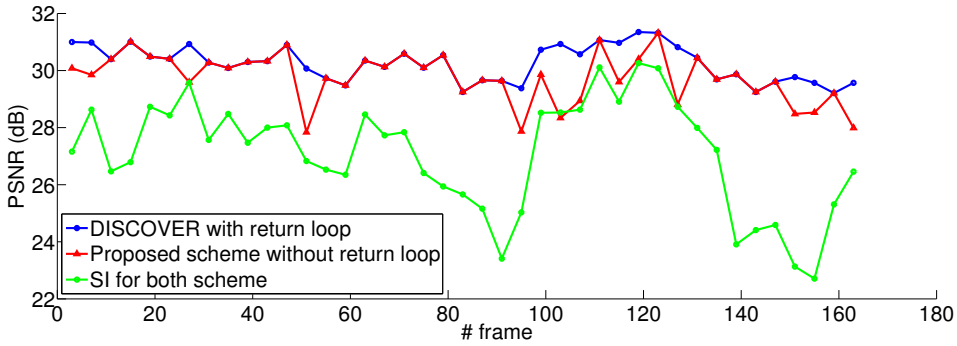
sensibly damage the performances. More precisely, the bit error probability evolution (in function of the rate) can be very fast [Berrou, Glavieux, 1996]: even with a small rate underestimation, the error probability can be far greater than 10^{-3} (error value reached when the Discover optimal rate is sent). The PSNR difference can sometimes be around 3dB if only one bitplane is badly reconstructed. Obviously, damages are larger if the first bitplane is not well recovered rather than the last one, thus the security rate addition favors the first bitplanes.

3.3.3.3 Experiments

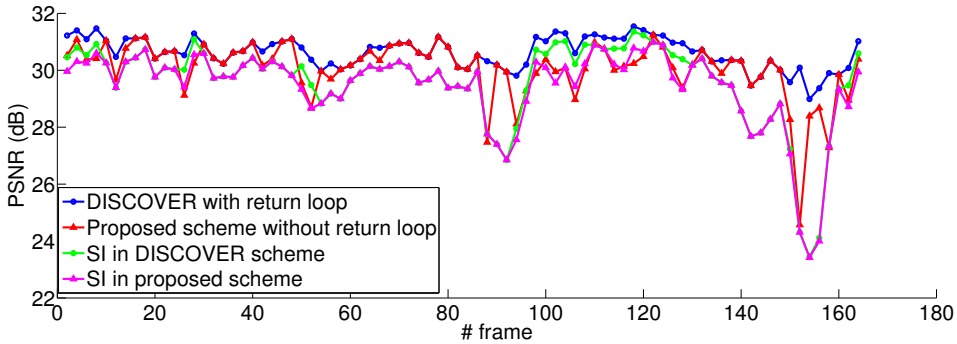
For several sequences, we tested the proposed bitplane rate estimation (based on the frame rate level estimation presented in Section 3.3.2). For each of them, we compare the average rates and the average PSNR of decoded frames. Results are presented in Table 3.4.
The obtained results show that the proposed approach degrades the optimal (but unattainable) Discover performance by $0.3-0.4$ dB and requires around 30% of additional bitrate. At first sight the results may seem disappointing, because of the sensible degradation of Discover efficiency. In fact, the performances of the proposed method are acceptable for the following reasons.

First, as already explained, the Discover scheme transmits the optimal rate and then such optimized performances should be seen as oracle results that any return-loop-free scheme would hardly achieve. A suppression of the return loop necessarily leads to a loss of video quality and/or an excess of transmitted rate.

Moreover, whereas it is difficult to precisely compare our results to the ones obtained by the existing methods (mainly because they use a GOP size of 2 for rate control), one can make several remarks anyway. Firstly, we can observe that for the scheme in pixel domain (Stanford scheme) proposed by Morbee *et al.* [Morbee *et al.*, 2007] the obtained losses offer a similar order of magnitude. For instance, for *foreman* sequence (with a GOP size of 2), their rate increase was around 40%, which is more than with our method. Secondly, Brites *et al.*. in [Brites, Pereira, 2007] have obtained an average loss of around 1.2 dB. Even if the experimental conditions are not the same, if we measure with the Bjontegaard metric [Bjontegaard, 2001] the gap between the Discover scheme and the proposed backward channel free algorithm (see Figure 3.14 for *suzie*), the loss is about 0.66 dB. If we cannot state precisely if our method outperforms the literature ones, we are able to affirm that our method works pretty well and leads to losses of the same order of magnitude as existing techniques do.



(a)



(b)

Figure 3.13: Comparison between the decoded frame PSNR for DISCOVER scheme (with a return loop) and for the proposed solution (without return loop) for the W_m (a) and W_l (b). *foreman* sequence (key frames at QP 40).

In Figure 3.13, we can see the PSNR evolution along the time of the decoded WZ frames, reconstructed with the proposed algorithm and with the reference DISCOVER scheme, and of the side information. One can remark that the losses are localized in some frames where the loss in magnitude can be more than $1 - 2$ dB. This is explained by the fact that the rates for these frames is underestimated and then the reconstruction quality strongly affected. Furthermore, one can see that when a middle WZ frame is badly estimated (Figure 3.13 (a)), the error propagates in the rest of the GOP (the lateral WZ frame, Figure 3.13 (b)).

The main drawback of the proposed technique is that it depends on several parameters estimated offline and which vary from the sequence (the μ coefficients, the multiplying factors to adjust the estimated rate to the theoretical rates and the security factors). This is obviously one major limit of our solution, which however remains promising, because of its encouraging results, and because it is conceivable to estimate these parameters online at the encoder, based on other available informations.

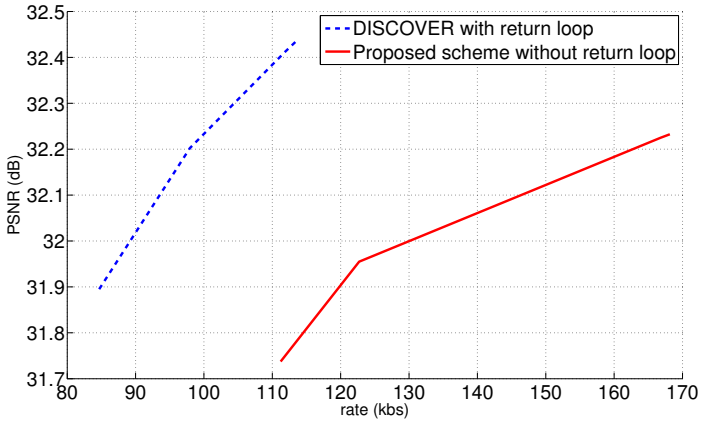
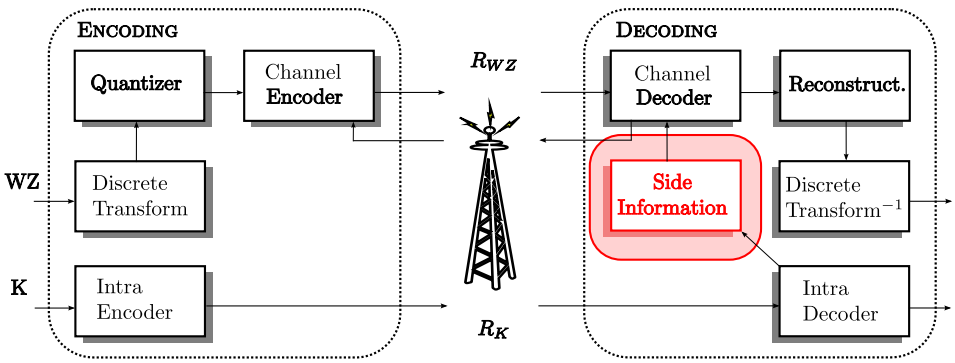


Figure 3.14: Rate-distortion comparison between Discover (with return-loop) and proposed scheme (without return loop), for *suzie* (QCIF, 176×144).

3.4 Conclusion

In this chapter, we studied three important issues of DVC. First, we have proposed a new frame repartition, less complex and more efficient than the ones existing in the literature and, based on the proposed distortion model, we have determined the optimal decoding order.

The second issue was the study of error propagation in the GOP in case of frame loss. Thanks to this analysis, we have confirmed that the different frames have not the same role and importance in the GOP. This observation lead us to look into the rate allocation between the frames. This was studied in the third part of this chapter, when we have proposed a rate estimation algorithm in order to get rid of the backward channel, one of the main drawbacks in DVC. Our technique presents interesting and promising results, but is still dependent on some parameters which need to be determined offline and which depend on the sequence.



Part II

Side information construction

"Distributed video coding performance strongly depends on the side information quality."

Chapter 4

State-of-the-art of the side information generation

In this chapter, we present the main existing types of side information generation methods, and for each of them, the main and more efficient techniques. This study will lead us to see several types of configuration depending on monoview/muliview settings, frame classification, available reference frame, available context information (depth, scene, etc.)...

First, in Section 4.1 we present the methods used for generating an estimation of the WZ frame, and then, in Section 4.2, we will study the case when there are several available estimations which need to be merged pixel by pixel. Finally, in Section 4.3, we describe the hash-based schemes designed for transmitting some localized and well-chosen WZ information, in order to help the side information generation process at the decoder.

Contents

4.1 Estimation methods 109
4.1.1 Interpolation 109
4.1.2 Extrapolation 114
4.1.3 Disparity 117
4.1.4 Spatial estimation 119
4.1.5 Refinement methods 120
4.2 Fusion 121
4.2.1 Problem statement 121
4.2.2 Symmetric schemes 121
4.2.3 Other schemes 123
4.3 Hash-based schemes 124
4.3.1 Definition of a hash-based scheme 124
4.3.2 Hash information transmission 124
4.3.3 Hash based side information generation methods 126
4.4 Conclusion 126

Distributed video coding performances do not achieve yet the classical inter frame video coding scheme ones, as they ideally could. One of the reasons is arguably that the quality of the side information is not yet good enough. Indeed, at the decoder side, the turbocodes or LDPC, correct the side information while using parity information sent by the encoder. If the correlation noise model is determined and not far from the true error distribution (see Chapter 8 for more details), the more precise the Wyner-Ziv estimation (closed to the original WZ frame), the less bits would be required for the SI correction by the channel decoder. Thus, many works have been conducted in order to build a more precise WZ estimation, by exploiting several kinds of available information (already decoded frames, geometry of the scene in case of multiview coding, etc.).

In this chapter, we propose a review of the main existing side information generation algorithms. They differ in their complexity but also from the point of view of the schemes they are based on. Indeed, a method developed for a multiview configuration has not the same issues as those designed for monoview video coding or even for stereo coding. They also depend on the frame distribution (GOP size, frame type disposition in the time-view space for multiview coding). Some works propose a review of the literature but they are limited to one configuration. For example, in [Artigas *et al.*, 2007b], Artigas *et al.*. describe some of the existing methods for multiview coding, but only for a special frame distribution in the time-view space (which we called hybrid scheme). Though we expose here the methods for several configurations, we will only give the algorithms which are based on the Stanford scheme and not those based on the PRISM approach.

Distributed video coding aims at reducing the encoding complexity while shifting the inter frame estimation to the decoder. Then the major part of the existing side information generation algorithms does not deal with the computation time issue since estimation is performed at the decoder side, where the computational capacity is assumed to be very powerful. However, some works, as that of Wang *et al.*. in [Wang, Liu, 2009], propose a parallel implementation of a side information generation method, which is then faster. But finally, knowing that the iterative channel decoder is far more complex than the usual WZ estimation methods, and knowing that the general DVC scheme is still suboptimal nowadays, it is probably a little too early, quite unuseful and hopeless to set the purpose of reducing the algorithms complexity.

In the following we will adopt these following notations: the original estimated WZ frame belonging to the nth camera ($n \in \mathbb{N}$) at time t ($t \in \mathbb{N}$) is denoted by W, and its generated side information by $\widehat{W}$. $\hat{I}_{m,k}$ denotes the already decoded reference frame which is the kth frame of the mth camera. In case of monoview estimations, the notation $\hat{I}_{0,k}$ is simplified to $\hat{I}_k$. In other words, when the reference frames have only one index, it means by default that we are in the case of monoview coding.

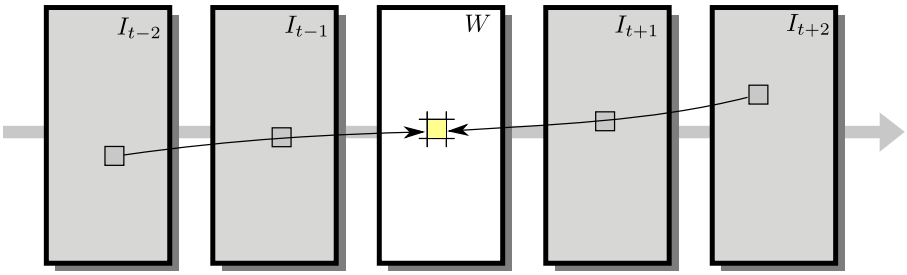


Figure 4.1: Interpolation methods for side information generation use already decoded frames which are before and after, or left and right, the estimated WZ frame.

4.1 Estimation methods

4.1.1 Interpolation

The mathematical interpolation concept consists in estimating an unknown information from other available neighbouring informations. Thus, as presented in Figure 4.1, interpolation algorithms in DVC are based on reference frames or in general on already decoded frames (because reconstructed WZ frames can also be used) which are before and after, or left and right, the WZ frame to be estimated, W.

The simplest interpolation is the frame averaging and was used at the very beginning of DVC [Aaron *et al.*, 2002]. For every pixel $\mathbf{p} \in [\![1, N_{\text{height}}]\!] \times [\![1, N_{\text{width}}]\!]$, the WZ frame estimation is the average of the two neighboring frames, I_{t-1} and I_{t+1}, pixel values[1]:

$$W(\mathbf{p}) = \frac{1}{2}\Big(I_{t-1}(\mathbf{p}) + I_{t+1}(\mathbf{p})\Big).$$

This very naïve method is not complex at all, and moreover, it can be very efficient in case of low motion (for instance, the beginning of the video *hall monitor*in Figure 4.4 (a)). On the contrary, average based interpolation leads to a very poor side information when motion activity is more intense (Figures 4.4 (b) and (c)).

As a consequence, the techniques proposed afterwards were more sophisticated and efficient, since they take into account the motion of the scene. They are called *motion interpolation (MI) methods*, and constitute the main category of the existing types of SI generation algorithms. They consist in estimating the two motion vector fields, $\mathbf{u}_{t-1}$ and $\mathbf{u}_{t+1}$, respectively between W and I_{t-1}, and W and I_{t+1}, and after in averaging the two compensated frames, $\forall \mathbf{p} \in [\![1, N_{\text{height}}]\!] \times [\![1, N_{\text{width}}]\!]$:

$$W(\mathbf{p}) = \frac{1}{2}\Big(I_{t-1}(\mathbf{p} - \mathbf{u}_{t-1}(\mathbf{p})) + I_{t+1}(\mathbf{p} - \mathbf{u}_{t+1}(\mathbf{p}))\Big).$$

The first MI technique is the simplest one and was also proposed at the beginning of DVC [Aaron *et al.*, 2002]. It is called *symmetric motion vector (SMV) interpolation*, and is a naive bidirectional motion estimation. The motion vectors are obtained, block per block,

[1]We adopt for this formula and for some others in the following the monoview notation because it has initially risen from works dealing with the temporal direction, but it can be easily extended to the multicamera case.

by finding the best symmetric motion vector fields (*i.e.*,symmetric means $\mathbf{u}_{t-1} = -\mathbf{u}_{t+1}$). This estimation of the best candidate for a block, $\mathbf{b}$, is performed by calculating for each tested vector, $\mathbf{u}_{tested}$ the following sum square distance (SSD):

$$SSD = \sum_{\mathbf{p}\in\mathbf{b}} \Big(I_{t-1}(\mathbf{p} - \mathbf{u}_{tested}(\mathbf{p})) - I_{t+1}(\mathbf{p} + \mathbf{u}_{tested}(\mathbf{p})) \Big)^2.$$

The chosen vector is the one which achieves the lowest SSD, assuming the hypothesis that the motion vector estimation is good when the forward estimation is similar to the backward estimation. Another hypothesis is that the motion is completely linear and symmetric. This method is quite efficient, and better than the average when motion activity is present (see Figures 4.4 (d) and (e)). But it is however not robust when motion is complex (see Figure 4.4 (f)). However, it was commonly used in the DVC literature, as in [Girod *et al.*, 2005][Guo *et al.*, 2006a][Ouaret *et al.*, 2006][Ouaret *et al.*, 2007][Yaacoub *et al.*, 2009a]. Aaron *et al.*. used it in the case of GOP size equal to 4, [Aaron *et al.*, 2003], proposing to use a hierarchical structure in the WZ frame decoding order (see Section 1.2.2.2.a).

Aware of the fact that the simple SMV method can be rapidly limited in case of complex motion, several works have been done in order to enhance this technique and make it more sophisticated. Some of them were inspired by interpolation algorithms developed outside of the DVC framework, for example by Zhai [Zhai *et al.*, 2005], or by Chen in 2002 [Chen, 2002], who performed two motion estimations: a forward (between I_{t-1} and I_{t+1}) and a backward (between I_{t-1} and I_{t+1}) one. Then, the obtained motion vectors are divided by two, and finally the two estimations are merged while choosing block per block the best estimation. This method is called *motion compensated frame interpolation (MCFI)*.
In 2004, Aaron *et al.*. [Aaron *et al.*, 2004b] improved their initial SMV method by adding to the bidirectional block matching, smoothness constraints on the estimated motion, and perform an overlapped block motion compensation (in case of GOP size of 2).
In [Artigas *et al.*, 2006], Artigas *et al.*. use a technique proposed by Lee *et al.*. in [Lee *et al.*, 2003] for the purpose of frame up-conversion in the classical coding, which presents several similarities with the issues involved in the side information generation for DVC.

Another improvement of the simple MCFI method is proposed by Dinh *et al.*. [Dinh *et al.*, 2007]. They use edge information to perform the motion estimation. Indeed, edges can help to define objects and then to define classes of vectors, because generally vectors are identical inside an object. In general, it is interesting to take into account the geometry of the scene in an interpolation method. If algorithms only take into account the SSD or SAD (sum of the absolute differences) between compensated blocks, they can sometimes match a very similar block in the other image (and choose the corresponding vector), but which does not correspond to the same object than the initial block. This is not a matter in a classical estimation/compensation problem, because the goal is only to minimise the mean square error. But in order to perform an interpolation, when dividing the motion vector by two to estimate the middle frame, the estimated vector does not necessarily correspond physically to the scene, and then the interpolation would not be precise.

The largest advance for side information generation was proposed by the members of the European project Discover [DISCOVER-website, 2005][Artigas *et al.*, 2007a] (DIS-

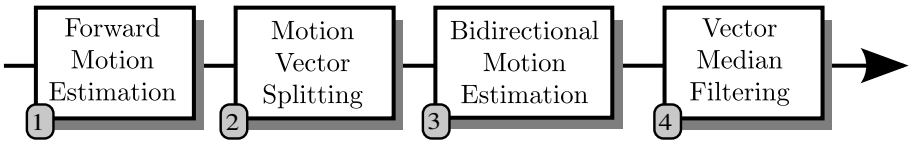


Figure 4.2: DISCOVER interpolation method.

tributed COding for Video sERvices). The elaborated technique [Ascenso *et al.*, 2005a] is nowadays the most popular algorithm and other researches compare their performance to it. This is why we present here in detail this method, schematized in Figure 4.2.

The DISCOVER method is constituted by four steps. The input of this algorithm are the two reference frames I_{k_1} and I_{k_2}, and the output are the two motion vector fields $\mathbf{u}_{k_1}$ and $\mathbf{u}_{k_2}$. The following is the detail of each block.

1. *Forward motion estimation* - the algorithm starts with a motion estimation between the two reference frames. For each block of I_{k_2}, the vector which points onto the most similar block of I_{k_1} is found. Let $\mathbf{b}_{k_1}$ and $\mathbf{b}_{k_2}$ be two blocks of respectively I_{k_1} and I_{k_2}, related to a vector $\mathbf{u}$. The similarity between them is calculated with the following criterion called *weighted mean absolute difference (WMAD)*:

$$WMAD(\mathbf{b}_{k_1}, \mathbf{b}_{k_2}) = \frac{1}{N_b} \sum_{\mathbf{p} \in \mathbf{b}_{k_2}} \Big| I_{k_2}(\mathbf{p}) - I_{k_1}(\mathbf{p} - \mathbf{u}) \Big| \Big(1 + \lambda \|\mathbf{u}\|_2 \Big) \tag{4.1}$$

   where N_b is the block size. This criterion is a classical mean absolute difference (MAD), with an regularization additional term $\lambda\|\mathbf{u}\|_2$ which penalizes large vectors. This criterion is crucial and is one of the reasons why DISCOVER obtains very good performances. For some images in some sequences, the difference between WMAD and MAD can achieve 2 dB. The experimental optimal value for λ is 0.05 [Ascenso *et al.*, 2006].

2. *Motion vector splitting* - the second step of DISCOVER algorithm consists in establishing for each block of the WZ frame W a bidirectional vector determined from the vectors calculated in the first step (see Figure 4.3). This is done by:

   - firstly dividing by two the vectors of the forward motion estimation
   - then selecting the best motion vector for each block. In other words, for each block, the algorithm chooses among the half forward motion vector, the one which points to W the closest to the centre of the block. Then, this selected vector is shifted to the centre of the block, and extended by symmetry, in order to obtain a bidirectional motion vector.

3. *Bidirectional motion estimation* - the next step is a simple bidirectional motion estimation around the initial position determined previously. The best vector is chosen by minimizing the WMAD metric as in step 1 (Equation (4.1)), slightly modified to be adapted to the bidirectional mode:

$$WMAD(\mathbf{b}_{k_1}, \mathbf{b}_{k_2}) = \frac{1}{N_b} \sum_{\mathbf{p} \in \mathbf{b}_{k_2}} \Big| I_{k_2}(\mathbf{p} + \mathbf{u}) - I_{k_1}(\mathbf{p} - \mathbf{u}) \Big| \Big(1 + \lambda \|\mathbf{u}\|_2 \Big) \tag{4.2}$$

   with the same hypotheses as in Equation (4.1).

4. *Vector median filtering* - at this stage, the motion vectors often present small spatial incoherences, and then need to be smoothed. The Discover method proposes to use a weighted median filter as in [Alparone *et al.*, 1996]. The filtered vector $\mathbf{u}_{fil}$ is:

$$\mathbf{u}_{fil} = \min_{\mathbf{u}} \sum_{j=1}^{N_{neighbour}} w_j \|\mathbf{u} - \mathbf{u}_j\|_1 \tag{4.3}$$

with

$$w_j = \sum_{\mathbf{p} \in \mathbf{b}_{k_2}} |I_{k_2}(\mathbf{p} + \mathbf{u}_j) - I_{k_1}(\mathbf{p} - \mathbf{u}_j)|^2$$

and where $\mathbf{u}_j$ are the neighboring vectors. The obtained vector is then close to its reliable neighbours.

This method was proposed in a pixel-domain context, but was also commonly used and competitive in transform-domain schemes [Brites *et al.*, 2006b]. Moreover, the Discover algorithm was designed for estimating a WZ frame between two key frames which are placed directly before and after it, in other words, in a GOP size of 2. But in [Ascenso *et al.*, 2006], Ascenso *et al.* proposed a flexible GOP size scheme. As a consequence, the Discover technique is used for long-term estimations, and also for non-symmetric interpolations, *i.e.*,when the distance with the backward frame is different from the distance with the forward frame. There is no major modification to obtain such asymmetric interpolations. Indeed, we only need to divide the motion vector by the appropriate value (instead of 2). In addition, Ascenso *et al.*. add a second bidirectional estimation, just after the first one, but with a finer block size (half width and height) and with a smaller search window. This additional step achieves a $0.1 - 0.2$ dB gain compared to [Ascenso *et al.*, 2005a] initial technique.

Several other Discover improvements have been proposed in the literature, as those by Klomb *et al.*, [Klomp *et al.*, 2006], who developed a similar technique involving sub-pel estimation. In [Huang, Forchhammer, 2008], Huang *et al.* complete the scheme in Figure 4.2 with two additionally blocks and by performing the technique with the three Y, U and V components. The first one is another bidirectional motion estimation but this time the block size is variable. Then, for the final construction step, the classical average of the two motion compensations is replaced by an overlapped block motion compensation (OBMC), as in [Lee *et al.*, 2003]. In practice, the most sensible improvement due to these techniques is the OBMC. The U and V information utilization does not change sensibly the SI quality, and besides, the variable block search does not lead to large gains. More recently, Ascenso and Pereira [Ascenso, Pereira, 2008] proposed a clear description of every block of Discover technique, and its possible refinements.

The previous methods adopt a block-based approach for frame interpolation. In other words, the motion is estimated by blocks of diverse sizes. Some other works have chosen a different approach, like Kubasov *et al.*. in [Kubasov, Guillemot, 2006] who propose to estimate the motion based on a triangularization of the reference frames. First, the reference frame I_{t-1} is meshed. Then they perform an estimation of the mesh position in the reference frame I_{t+1}. At the end, they perform the interpolation based on this mesh displacement estimation. This original approach does not bring by itself sensible benefits, thus they proposed to make a hybrid estimation: block based merged with mesh-based

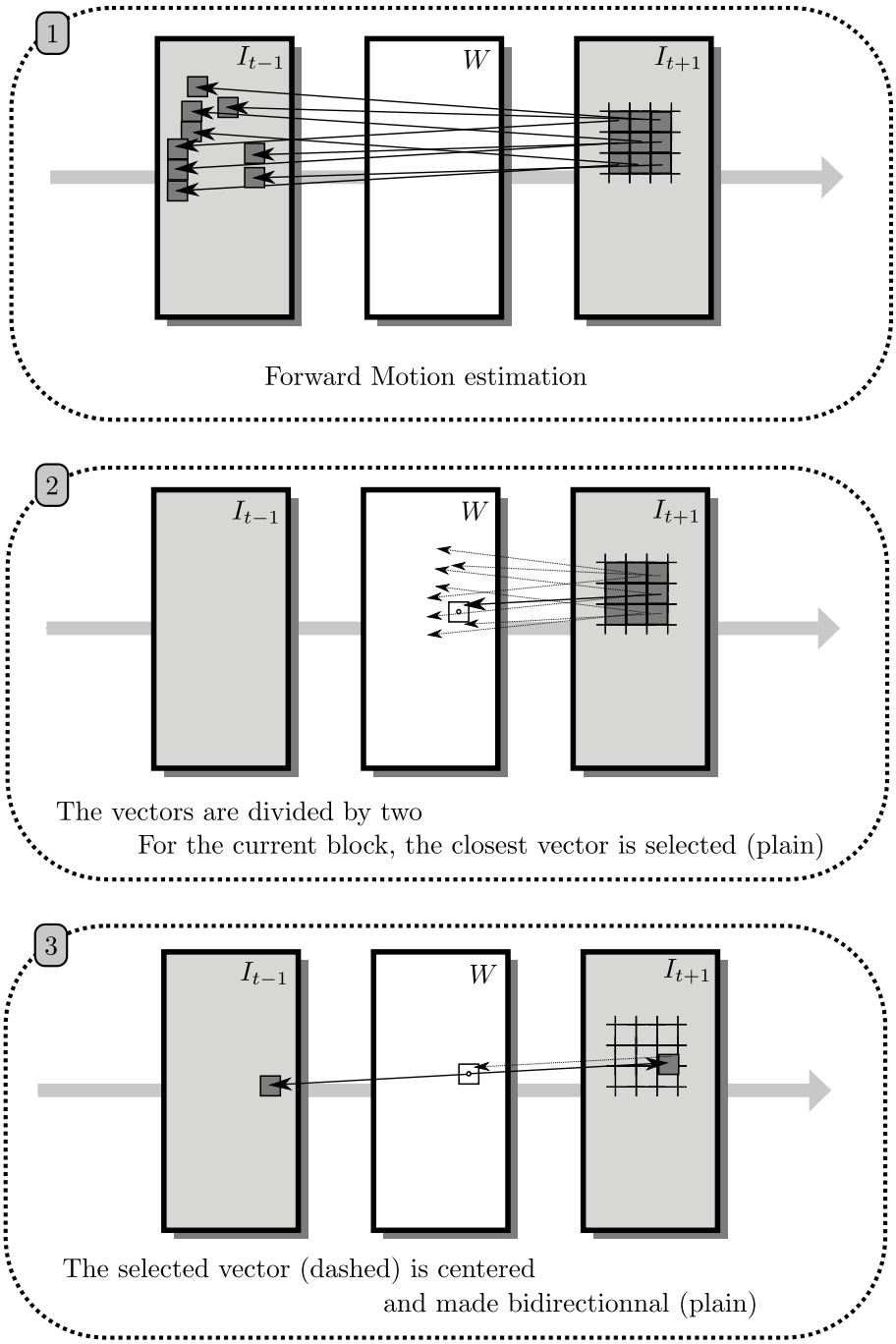


Figure 4.3: DISCOVER vector splitting.

(a) Average 39.43 dB.

(b) Average 26.15 dB.

(c) Average 24.28 dB.

(d) SMV 39.36 dB.

(e) SMV 28.98 dB.

(f) SMV 27.00 dB.

(g) Discover 39.38 dB.

(h) Discover 29.51 dB.

(i) Discover 29.03 dB.

Figure 4.4: (a),(d) and (g) *hall monitor*sequence with no motion: all of the methods obtain the same SI quality - (b), (e) and (h) *coastguard*sequence with a linear background motion: Average method fails while both motion based techniques construct a equivalent quality SI - (c), (f) and (i) *foreman*sequence with a complex motion: Average and SMV fail, and only Discover obtains an acceptable SI

interpolation. The results show that if we perform a ideal fusion (oracle) of the two estimations (mesh-based and block-based), the gain can be acceptable (around 1 dB), but with a real and feasible fusion, the gain is low.

Another novelty for frame interpolation in the monoview DVC framework is to use more than 2 reference frames. Recently, Petrazzuoli *et al.*. [Petrazzuoli *et al.*, 2010] proposed to use 4 reference frames I_{t-3}, I_{t-1}, I_{t+1} and I_{t+3} in order to obtain a non-linear interpolated trajectory and then model more complex motions. The gain obtained by this method are promising and show the need of considering more complex motion models for further improvement in side information generation.

4.1.2 Extrapolation

Interpolation techniques give, for the most recent of them, side informations of quite acceptable quality. However, they obtain good results only for limited GOP size (as 2 or

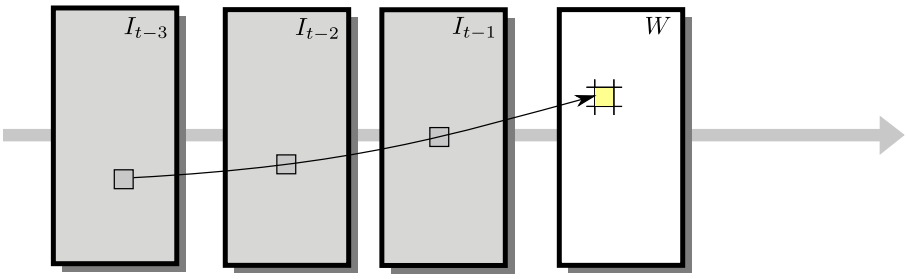


Figure 4.5: Extrapolation methods for side information generation use past decoded frames which are before the estimated WZ frame.

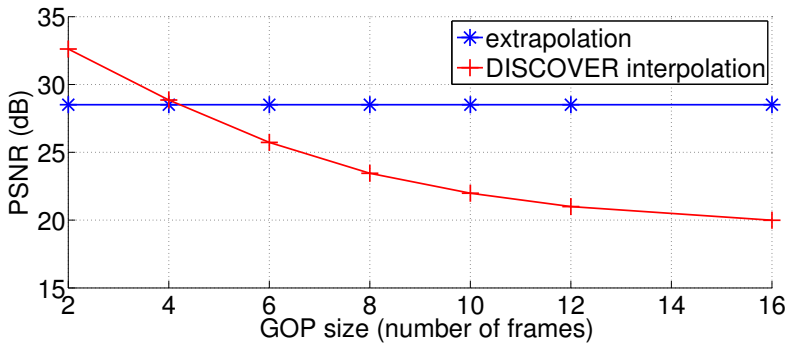


Figure 4.6: Comparison of side information PSNR between extrapolation and interpolation techniques for increasing GOP size, for *foreman*sequence.

4, but rarely more), because, by definition, interpolation algorithms need one reference frame before, and one after the estimated WZ frame. If the two reference frames are too far away (with more than 3 frames between them), the interpolation quality is sensibly degraded. This explains the fact that *extrapolation* methods have been proposed, because they only use the past information, *i.e.*,the past already decoded frame(s) (see Figure 4.5). In general, extrapolation techniques give a less precise estimation than the interpolation, but they are more convenient because they can be used with longer size blocks (8, 16, 32 and even more), without decreasing the quality. It is explained in [Tagliasacchi *et al.*, 2006b], and we present in Figure 4.6, some tests for *foreman* sequence, which show the evolution of estimation quality of interpolation and extrapolation, while the GOP size is increasing. One can observe in Figure 4.6 that whenever the GOP size is greater than 4, interpolation performance becomes lower than extrapolation one.

The first motion extrapolation methods were introduced in [Aaron *et al.*, 2004b][Girod *et al.*, 2005]: the *motion compensated extrapolation (MCE)*. The principle is simple. Let us assume that two frames are available at the decoder (*i.e.*,they are already decoded). These two frames are just before the estimated WZ frame W. If W is at time t, we denote the two extrapolated frames I_{t-2} and I_{t-1}.
The method consists in firstly estimating the motion between I_{t-2} and I_{t-1} (*e.g.* by block matching with smoothness constraint). Then, the motion is extrapolated to time t and the side information is constructed by calculating the overlapped motion compensation of frame I_{t-1}. This non complex technique is not very competitive and does not achieve the motion interpolation performance for the case of short GOP. On the contrary, when the

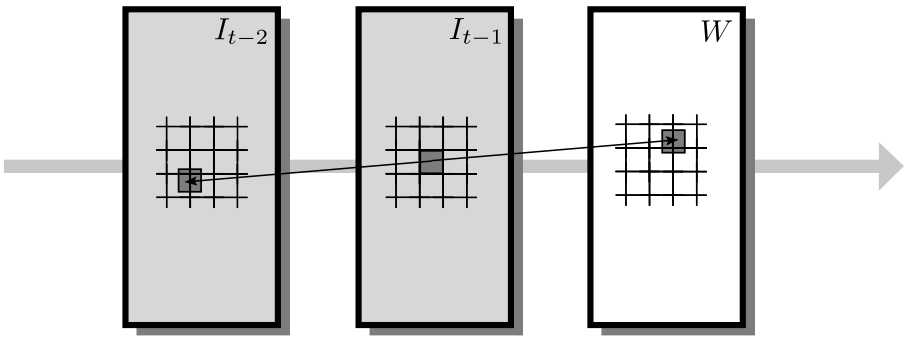


Figure 4.7: The motion vectors between I_{t-1} and I_{t-2} are used for extrapolating the frame W.

GOP size is large, the interpolation efficiency is degraded and at this moment, the simple MCE offers a better description of the WZ frame.

In 2005, Natario *et al.*. [Natario *et al.*, 2005] give a detailed version of the MCE algorithm, which can be decomposed in 4 steps:

- **Motion estimation** between I_{t-2} and I_{t-1}. As shown in Figure 4.7, for each block of I_{t-1}, a block matching is performed in order to find the most similar block in I_{t-2}, in order to obtain the motion vector field $\mathbf{u}$.

- **Motion field filtering** which consists in averaging the vectors $\mathbf{u}$ with their neighbours in order to obtain a smoothed field, enabling to construct a better side information.

- **Motion projection** from frame I_{t-1} to frame W. More precisely, for each block $\mathbf{b}$, a vector $\mathbf{u_b}$ was computed in step 1 and the projection consists stating for block $\mathbf{b}$, the vector $-\mathbf{u_b}$ as the motion vector between I_{t-1} and W.

- **Overlapping and uncovered areas**. After motion compensation (with the $-\mathbf{u_b}$ motion vector field), some pixels would be estimated by more than one candidate (coming from different blocks). In this case, an average of the estimation values is performed. On the contrary, it happens that some areas of the WZ frame are not covered by the compensated blocks. In this case a rapid spatial estimation is performed (average of the available neighbours).

Borchert *et al.*. in [Borchert *et al.*, 2007a][Borchert *et al.*, 2007b] propose a more sophisticated extrapolation technique. Instead of using 2 reference frames, their algorithm is based on three frames I_{t-3}, I_{t-2} and I_{t-1}. They perform several motion estimations: between I_{t-2} and I_{t-3}, between I_{t-2} and I_{t-1} and between I_{t-1} and I_{t-2}. Based on the estimated vector fields they can predict the uncovered areas and perform temporal hole-filling. They obtain a sensible gain compared to simple MCE, especially where motion is more complex (for example with *carphone* sequence).

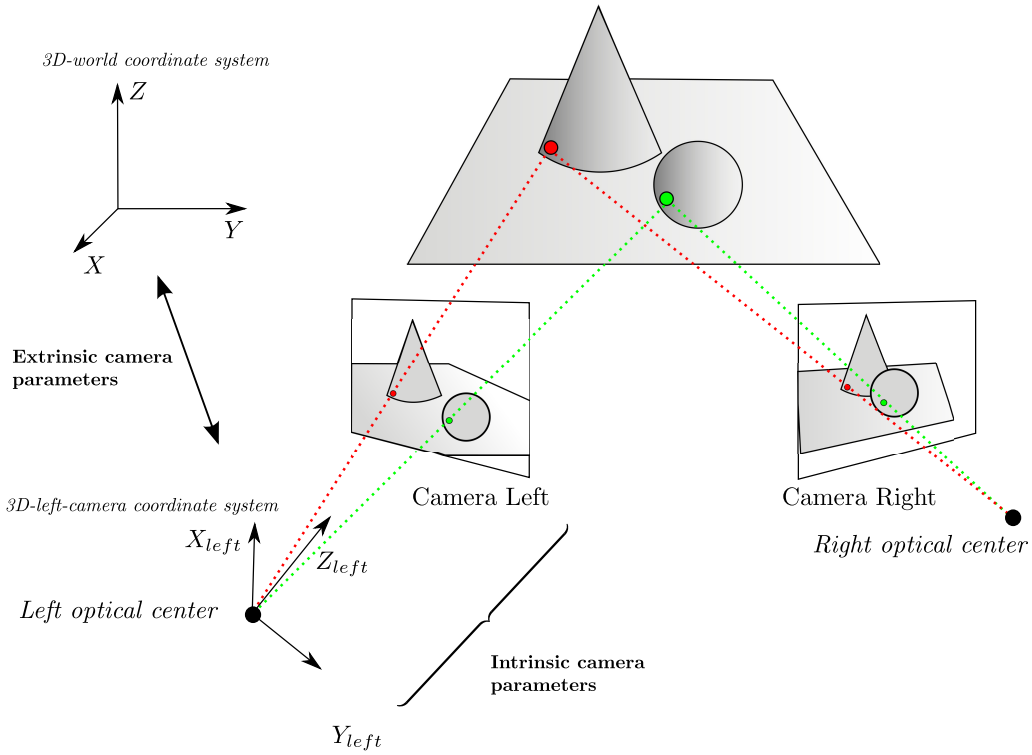


Figure 4.8: 3D-scene acquisition by two stereo cameras, with their own spatial centre and coordinate system.

4.1.3 Disparity

Though the techniques presented before can be used in a multiview context, they were designed at the origin for side information construction of monoview sequences. In other words, they were destined to perform motion interpolation, and to model motion activity. In the view direction, the difference between frames (same time but coming from different cameras) is called *disparity* and does not present exactly the same properties as motion. Indeed, in motion estimation, the purpose is to detect the background and objects in the scene, and to find their movement. For disparity estimation, once the objects and the background determined, the goal is to detect their displacement (depending on the depth), but also their deformation, due to camera disposition.

A clear and detailed description of the 3D geometry and the camera projection problem is proposed in the PhD thesis of Daribo [Daribo, 2009]. The different elements of these issues are summarized in Figure 4.8. A 3D scene acquisition is performed by two cameras (left and right). Each of the cameras has its own intrinsic parameters. Disparity estimation techniques must take into account these elements while calculating the disparity field. They also need to use the extrinsic parameters which correspond to every kind of external information (as the position and the orientation). At the end, once the camera parameters are known, the 3D geometry based methods often require the depth information (the depth corresponds to the distance between the object and the camera optical center). More precisely, when at the left camera, one is able to know for each point of the image the distance, d, between the camera plane and the true point in the 3D scene, it is possible

to know easily how to project it on the right camera. In the particular case of rectified cameras (*i.e.*,when all the points on a line of left image are set on the same line in the right image) the link between depth and projection is very simple. Every point of the left image is translated horizontally (with a vector $\mathbf{u}$), proportionally to the inverse of its depth, d, following the relation:

$$\mathbf{u} = \frac{f \cdot t}{d}$$

where f is the focal distance, and t the distance between the two cameras.

Though the problematic of inter-view prediction is quite different from motion estimation, several works adopted the block-based motion interpolation techniques for disparity estimation. That is the case of Areia *et al.*. [Areia *et al.*, 2007] who use Discover algorithm (see Section 4.1.1) for inter camera disparity estimation. They justify this by the fact that disparity-based methods are not performant compared to Discover technique even if they better fit the problematic. Indeed, this is true for some sequences where deformations between views are small, and where block matching can thus be adapted. Ouaret *et al.*, in [Ouaret *et al.*, 2007][Ouaret *et al.*, 2006], also use MCE technique for inter-view interpolation, but complete it with geometry based algorithm, described below.

Pure inter-view interpolation algorithms are not only based on block-matching as motion interpolation are. Indeed they involve the integration of the geometry of the scene and base their estimation on a 3-dimensional representation as several works which have been proposed out of the distributed video coding framework, as [Martinian *et al.*, 2006], [Chen, Williams, 1993], [Shum, Kang, 2000]. The DVC interpolation techniques presented below are mainly based on these works.

One technique is called *homography projection*, and was used in [Guo *et al.*, 2006a], [Ouaret *et al.*, 2007][Ouaret *et al.*, 2006]. This approach relies on a 3×3 matrix (called homography) which relates one view to another one in the homogeneous coordinates system. This matrix has 9 parameters $(a_{ij})_{i=1...3,j=1...3}$ (where $a_{33} = 1$). Based on this matrix, each point, (x_1,y_1), of the image in the first view is mapped to a point, (x_2,y_2), in the second view with the following relation:

$$\lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{4.4}$$

where λ is a scale factor. The previous equation yields:

$$x_2 = \frac{a_{11}x_1 + a_{12}y_1 + a_{13}}{a_{31}x_1 + a_{32}y_1 + 1} \qquad \text{and} \qquad y_2 = \frac{a_{21}x_1 + a_{22}y_1 + a_{23}}{a_{31}x_1 + a_{32}y_1 + 1}. \tag{4.5}$$

Some particular transformations are obtained for some value combination. For example, the homography matrix describes a pure translation when the diagonal terms (a_{11} and a_{22}) are equal to 1, and $a_{12} = a_{21} = a_{31} = a_{32} = 0$. Another example is when $a_{13} = a_{23} = 0$, the λ scale factor is equal to 1 and we have an affine transformation. In the general case it is called a perspective transformation. The parameters are computed using a gradient descent applied to minimize a criterion composed by the mean square difference between the original image and the wrapped image. One can notice than homography

estimation is a similar problem with global motion estimation, where the purpose is to estimate the global displacement of the camera [Dufaux, Konrad, 2000] (translation, rotation, zoom, etc.). The main disadvantage of that approach is that the scene is assumed to be on a planar surface, which is rapidly non verified especially when there are objects at the foreground and at the background (ex: *outdoor* sequence).

This disadvantage of not separating objects and foreground is avoided by Artigas *et al.*. in [Artigas *et al.*, 2006]. Their method uses the information of a depth map to project back the elements of an image on a 3D scene and re-project them on the second image. This works correctly while the depth map information is precise, because it takes into account every object of the scene. This method is however very limited because it requires a depth map transmission (then, higher rate), and it also requires a depth map capture (because it cannot be estimated at the encoder), which is rarely possible in the classical simple inter-view installations (only while using specific hardware as z-cameras).

Areia *et al.*. in [Areia *et al.*, 2007] mention another technique (but do not adopt it and prefer a motion interpolation algorithm) which consists in estimating the disparity field on a past pair of frames (already decoded) and extrapolate it to the current WZ frame. Several similar techniques exist but they are very limited because they are based on a particular type of frame repartition in the time-view space, which generally employ too many intra coded frames (more complex and less efficient).

4.1.4 Spatial estimation

The SI generation methods seen until now are based on neighboring reference frames, from which are extracted some informations of movement, disparity, or any kind of correlation. Other approaches are not based on other reference frames. The main advantage is that this prevents any error propagation (*i.e.*,when a reference image is not entirely recovered, the error propagates into the frame using it for its SI generation, which is not the case for spatial estimation).

In his PhD thesis [Kubasov, 2008], Kubasov presents a very simple spatial estimation algorithm. The general idea amounts to an intra coding (with H.264 intra) of a filtered and downsampled version of the original frame. The image is upsampled at the decoder. The results are surprising because the PSNR of the spatial estimation are quite good but, after decoding they requires more rate for a lower decoding quality. This pointed out the limits of the PSNR metric for the estimation of the SI quality. The reader can see Chapter 9 for more details.

Tagliasacchi *et al.*. [Tagliasacchi *et al.*, 2006a] propose a more advanced technique which consists in dividing the image in macroblocks into two sets separated like on a chessboard. One set is decoded using only a temporal interpolation while the other uses besides a spatial estimation which is generated thanks to the already decoding neighboring macroblocks. This method leads to an impovement of 1 dB compared to the scheme involving only the temporal side information.

4.1.5 Refinement methods

All of the methods presented above aim at building a unique side information and at decoding it after construction. This approach presents however a disadvantage: for instance, an error in the estimation of the side information propagates along the bitplanes. More precisely, an error on the ith coefficient of the nth band would require parity bits from the turbodecoder for all of the bitplanes. Another example: a block estimation error would require parity bits for all the bands of all the affected coefficients. The refinement methods proposed to reconstruct the side information while the turbodecoding is being performed, and based on the already decoded information. They differ in their level of refinement (band, pixel, bitplane) and in their structure, but they all have the same purpose: using the already turbo decoded information to perform a side information refinement for the rest of turbodecoding.

Let us first make a tour of methods performing bitplane by bitplane refinement. Firstly, Ascenso *et al.* in 2005 [Ascenso *et al.*, 2005b] introduced a novel technique to continuously refine the motion vectors used for the side information interpolation while the WZ bitplanes are decoded. First a classical interpolation is done, and after the first bitplane decoding, the decoder detects the blocks where the initial side information differs from the decoded frame. For the selected block, the side information is reestimated by block matching using the decoded frame information. This improves the side information for the remaining bitplanes to be decoded, thus increasing the coding efficiency.

Later, Adikari *et al.*. [Adikari *et al.*, 2006] proposed another bitplane level refinement algorithm using luminance and chrominance information, which was rapidly improved by Weerakkody *et al.*. [Weerakkody *et al.*, 2007] who proposed a spatial-temporal refinement algorithm extending the Adikari's work to iteratively improve the initial side information obtained by motion extrapolation; this comprises interleaving the initial estimation for error detection and flagging, followed by de-interleaving and filling of the flagged bits with an alternate iterative use of spatial and temporal prediction techniques.

Although, in [Ascenso *et al.*, 2005b; Adikari *et al.*, 2006; Weerakkody *et al.*, 2007], the authors present high rate-distortion gains (up to 3 dB in some sequences), the performance results are obtained using lossless key frames at the decoder, which is an impractical video coding scenario and really impact, on the final rate-distortion results.

An other type of refinement was proposed by Varodayan *et al.*. [Varodayan *et al.*, 2008] who developed a method to update the motion field throughout the decoding process using the previously decoded frame as side information. This proposal uses an unsupervised method to learn the forward motion vectors based on expectation maximization. This method was only compared with JPEG performances, this does not give a reliable information about its efficiency.

Martins *et al.*. in [Martins *et al.*, 2009] proposed novel side information refinement technique with new approaches, notably for the choice of level refinement. While other existing techniques perform refinement after each received bitplane, Martins' technique proposed to reestimate the side information after the decoding of each band. The advantage of this technique is that it is less complex than the other while keeping the same results. A similar

band approach was proposed by Badem *et al.*. [Badem *et al.*, 2009] and by Macchiavello *et al.*. [Macchiavello, De Queiroz, 2007] for a scalable approach.

The previous methods perform side information refinement during the turbodecoding process. In the following, we present other refinement techniques which perform several turbo decoding steps and between each of them, the decoder reestimates the side information. A first one is proposed by Artigas *et al.*. in [Artigas, Torres, 2005] whose technique consists in constructing an interpolation and after a turbo decoding, refining it and redecoding it. The obtained results are not so good in the general case, but for first estimations of poor quality, the refinement technique can lead to sensible rate-distortion improvement.

A more advanced method was proposed in [Ye *et al.*, 2008] for the monoview case and [Ouaret *et al.*, 2009] for the multiview configuration. First the decoder performs a classical interpolation (DISCOVER) which is turbodecoded. After that, the decoder detects suspicious motion vectors and refines them. After an optimal motion compensation (not necessarily an average of the two compensated reference frames), the frame is turbodecoded again. They obtain 0.6 dB of gain compared to the classical transform-domain scheme for several QCIF sequences.

4.2 Fusion

4.2.1 Problem statement

The fusion problem springs up since in the multi-view DVC context, one ends up with having several different estimations of the current WZ frame in order to have only one side information to correct with the parity bits at the turbodecoder. While previously we have seen several ways of generating a frame estimation (interpolation, extrapolation, etc.), in this section, we study the case when there are several estimations for one WZ frame. The fusion methods strongly depend on the adopted configuration (the available frames, estimation methods, etc.). More precisely, the state-of-the-art methods were developed in two different contexts. The first one is in the symmetric schemes (Section 4.2.2). We give more details for this configuration because it is the one adopted in our work, and the presented methods will constitute our references for comparison with the literature later. The second configuration is the case of non-symmetric schemes, where the adopted frame distributions is not satisfying for us because of a too high number of key frames (see Section 4.2.3).

4.2.2 Symmetric schemes

In this section, we review the existing solutions for the fusion problem in the case of a quincunx frame distribution (symmetric scheme presented in Section 3.1.1), in which we have two estimations for W coming from the temporal and the inter-view interpolations. This is illustrated in Figure 4.9: motion estimation produces two motion vector fields, $\mathbf{u}_b$ and $\mathbf{u}_f$, which in turn are used to provide temporal estimations of W from $I_{n,t-1}$ and $I_{n,t+1}$. Therefore, we note with $\hat{I}_{n,t^-} = I_{n,t-1}(\mathbf{u}_b)$ the prediction obtained by compensating the image $I_{n,t-1}$ with vector $\mathbf{u}_b$. Likewise, we have $\hat{I}_{n,t^+} = I_{n,t+1}(\mathbf{u}_f)$. As far as disparity estimation is concerned, we note the disparity fields as $\mathbf{u}_l$ and $\mathbf{u}_r$ (which have quite different characteristics from motion vector fields), and the corresponding estimations as $\hat{I}_{n^-,t}$ and

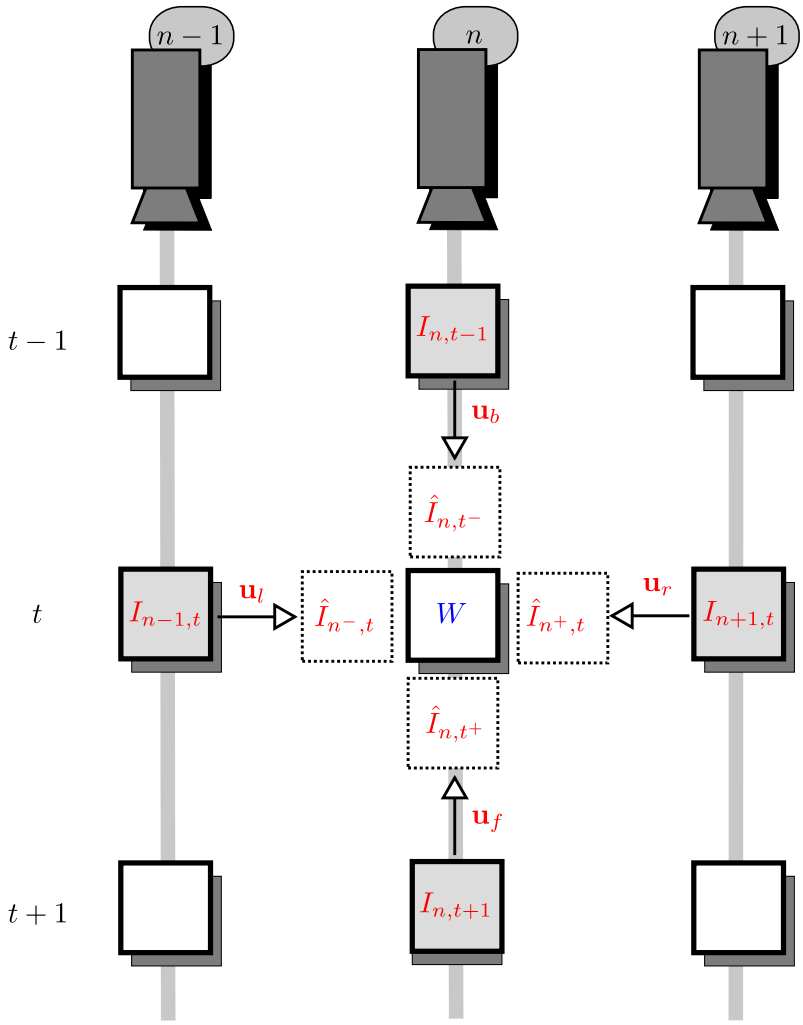


Figure 4.9: Fusion problem: I_x are the available KFs and $\hat{I}_x$ their motion compensated version, estimating the WZ frame W. $\mathbf{u}_x$ are the vector fields.

$\hat{I}_{n^+,t}$. Finally, the two temporal (or inter-view) estimations are combined in order to obtain a single estimation, respectively $\hat{I}_T = \frac{1}{2}\left(\hat{I}_{n,t^-} + \hat{I}_{n,t^+}\right)$ and $\hat{I}_N = \frac{1}{2}\left(\hat{I}_{n^-,t} + \hat{I}_{n^+,t}\right)$. The fusion problem amounts to produce an estimation of W from $\hat{I}_T$ and $\hat{I}_N$ with the target of minimizing the mean square error with respect to the actual WZ frame. In particular, an efficient fusion technique should produce a smaller MSE than both the individual estimations $\hat{I}_T$ and $\hat{I}_N$. All of the existing fusions are "binary" fusions, *i.e.*,pixel by pixel the merged value is taken from the temporal *or* the inter-view estimation.

The **ideal fusion** (Id), studied in [Areia *et al.*, 2007] is the upper bound one can achieve when performing a binary fusion. Pixel by pixel, the true estimation error, taking into account the original WZ frame, is computed and used as an oracle in order to decide what is the best value for the SI. The equation of the ideal fusion is for each pixel $\mathbf{p}$:

$$\widehat{W}(\mathbf{p}) = \begin{cases} \hat{I}_N(\mathbf{p}), & \text{if } |\hat{I}_N(\mathbf{p}) - W(\mathbf{p})| < |\hat{I}_T(\mathbf{p}) - W(\mathbf{p})| \\ \hat{I}_T(\mathbf{p}), & \text{otherwise.} \end{cases}$$

The **pixel difference fusion** (PD) was proposed by Ouaret et al. in [Ouaret *et al.*, 2006]. The interpolation error is estimated using the backward and forward frames of the same view. Two estimation errors are computed for the inter-view interpolation $E_N^b = |\hat{I}_N - I_{n,t-1}|$ and $E_N^f = |\hat{I}_N - I_{n,t+1}|$ and, similarly, for temporal interpolation $E_T^b = |\hat{I}_T - I_{n,t-1}|$ and $E_T^f = |\hat{I}_T - I_{n,t+1}|$. The equation of the PD fusion is therefore:

$$\widehat{W}(\mathbf{p}) = \begin{cases} \hat{I}_N(\mathbf{p}), & \text{if } E_N^b(\mathbf{p}) < E_T^b(\mathbf{p}) \text{ and } E_N^f(\mathbf{p}) < E_T^f(\mathbf{p}) \\ \hat{I}_T(\mathbf{p}), & \text{otherwise.} \end{cases}$$

The **motion compensated difference fusion** (MCD) was proposed by Guo *et al.*. in [Guo *et al.*, 2006a]. In this fusion algorithm, the absolute value of the difference between $\hat{I}_{n,t^-}$ and $\hat{I}_{n,t^+}$ is thresholded by T_1 and the motion vector values are also thresholded by T_2. The equation of the MCD fusion process is:

$$\widehat{W}(\mathbf{p}) = \begin{cases} \hat{I}_N(\mathbf{p}), & \text{if } |\hat{I}_{n,t^-}(\mathbf{p}) - \hat{I}_{n,t^+}(\mathbf{p})| > T_1 \\ & \quad \text{or } \|\mathbf{u}_b(\mathbf{p})\| > T_2 \\ & \quad \text{or } \|\mathbf{u}_f(\mathbf{p})\| > T_2 \\ \hat{I}_T(\mathbf{p}), & \text{otherwise.} \end{cases}$$

The **view projection fusion** (Vproj) was proposed by Ferré *et al.*. in [Ferre *et al.*, 2007]. In this case, the estimation $\hat{I}_T$ is projected onto $I_{n-1,t}$ and $I_{n+1,t}$. This projection consists in disparity compensations ($dc_l(\cdot)$ and $dc_r(\cdot)$) based on a simple block matching disparity estimation. The error images $E_l = I_{n-1,t} - dc_l(\hat{I}_T)$ and $E_r = I_{n+1,t} - dc_r(\hat{I}_T)$ are thresholded, leading to two masks which are projected back onto the WZ frame, with disparity compensations ($dc_l^{-1}(\cdot)$ and $dc_r^{-1}(\cdot)$) based on $\mathbf{u}_r$ and $\mathbf{u}_l$. The equation of the Vproj fusion process is:

$$\widehat{W}(\mathbf{p}) = \begin{cases} \hat{I}_N(\mathbf{p}), & \text{if } |dc_l^{-1}(E_l)(\mathbf{p})| > T \text{ or } |dc_r^{-1}(E_r)(\mathbf{p})| > T \\ \hat{I}_T(\mathbf{p}), & \text{otherwise.} \end{cases}$$

The **temporal projection fusion** (Tproj) was proposed by Ferré *et al.*. in [Ferre *et al.*, 2007]. It is the equivalent of the Vproj fusion in the temporal direction. The estimation $\hat{I}_N$ is first projected on $I_{n,t-1}$ and $I_{n,t+1}$ by motion compensation. Two error images, $E_b = I_{n,t-1} - mc_l(\hat{I}_N)$ and $E_f = I_{n,t+1} - mc_r(\hat{I}_N)$, are then thresholded and the obtained masks are projected back onto the original position. The equation of the Tproj fusion process is:

$$\widehat{W}(\mathbf{p}) = \begin{cases} \hat{I}_N(\mathbf{p}), & \text{if } mc_b^{-1}(E_b) < T \text{ or } mc_f^{-1}(E_f) < T \\ \hat{I}_T(\mathbf{p}), & \text{otherwise.} \end{cases}$$

4.2.3 Other schemes

As it was explained in the introduction of this section, fusion algorithms strongly depend on the adopted scheme. Some methods developed in a specific kind of frame distribution would not have the same initial hypotheses as the one presented above, based on a symmetric scheme. For example, several techniques are proposed based on hybrid schemes (see Section 3.1.1 for more details), *i.e.*,when the camera type alternates between complete intra, and mixed intra-WZ frames. In other words, for the estimation of a WZ frame, all the available frames directly "around" it are intra coded. This allows easier fusion based

on more numerous available informations.

In such a scheme, Artigas *et al.*. in [Artigas *et al.*, 2006][Artigas *et al.*, 2007b], proposed to use the fact that in the neighboring views, all the frames are known (because they are intra coded). In other words, the decoder calculates the temporal interpolation error in this view, and projects this image error to the current view, in order to use this information for the fusion. This method and others in the same spirit are interesting for their good utilization of the multiview aspect (projection on the neighboring views), but they need too much information at the decoder. Indeed, for one WZ frame, these methods need between 6 and 8 key frames, contrary to symmetric schemes based methods who need only 4 frames. This is why we do not detail these methods and we will not compare against them in the following.

4.3 Hash-based schemes

4.3.1 Definition of a hash-based scheme

In the classical distributed video schemes, the WZ transmission is done only through the Slepian-Wolf coder, in order to correct the side information generated at the decoder. The correlation between the side information and the original frame is not the same in all the frame. Some regions are badly estimated and would request a high number of parity bits, but others are well recovered and would require a lower rate. Moreover, at the decoder, while side information is generated, some regions cannot be estimated because they do not exist in the reference frames. For all of these reasons, some works propose to transmit some pieces of WZ information in order to enhance the side information estimation process. The issue of this problem is twofold, first, the specific WZ information, called *hash*, has to be selected and well chosen, and secondly, it has to be compressed and transmitted to the decoder. Then, at the decoder, the side information method uses the hash for a better estimation. The general structure of a hash-based DVC scheme is presented in Figure 4.10. Each of the blocks in bold (specific to hash-based schemes) are presented in detail in the following subsections. In the following, the key frame rate is given by R_K, the hash rate by R_H and the WZ parity bits rate by $R_{WZ}(R_H)$ (which depends on R_H). The objective of a hash-based scheme is, for a equivalent decoding quality, to obtain a WZ rate $(R_H + R_{WZ}(R_H))$ lower than the parity rate in case of no hash transmission, $R_{WZ}(0)$.

4.3.2 Hash information transmission

4.3.2.1 Hash selection

Hash information transmitted to the decoder aims at improving side information quality in some regions hard to estimate by classical algorithms (occlusions, rapid motion, etc.). Then, the encoder has to foresee the regions of the image to transmit, *i.e.*,the encoder has to guess where the interpolation at the decoder would fail. Indeed, easily estimated regions would not need hash information, and would thus reduce the rate R_H. Yaacoub *et al.*. in [Yaacoub *et al.*, 2009a; Yaacoub *et al.*, 2009b; Yaacoub *et al.*, 2009c] do not perform a hash selection: they transmit all of the 16×16 blocks, because their purpose is to measure the efficiency of their genetic algorithm in a hash based scheme and not to prove that a transmitting hash improves rate-distortion performances. In Chapter 7, we extend their

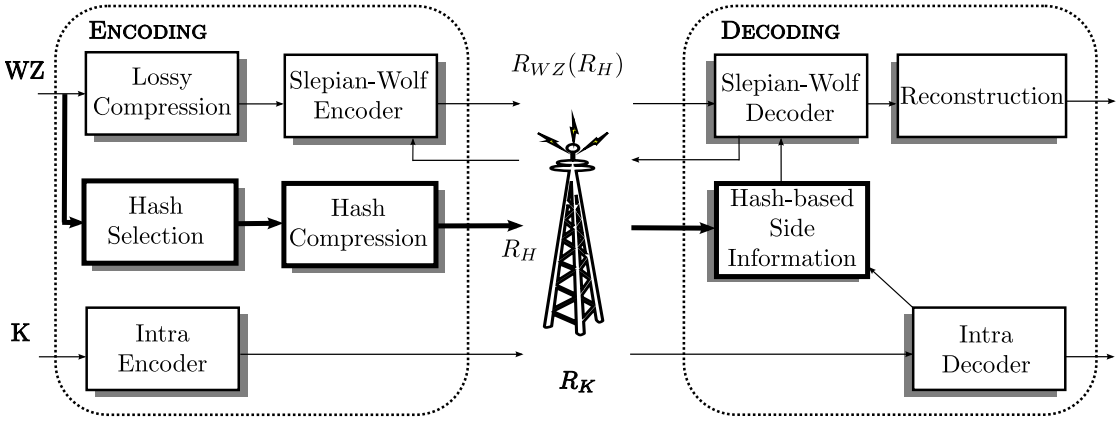


Figure 4.10: General structure of a Hash-based DVC scheme. The block with bold strokes are specific.

work and perform a block selection at the encoder.
Two Stanford-inspired hash based schemes were proposed by Aaron *et al.*. [Aaron *et al.*, 2004a], at the begining of DVC, and by [Ascenso, Pereira, 2007] in the context of DISCOVER project. Though they differ for the hash compression and for the proposed hash based side information generation techniques, their block selection is identical: the encoder thresholds the difference between the previous reference frame and the current frame for each macroblock. The hash information is sent only in the case where the sum square difference is greater than the threshold. In spite of the fact that using the previous key frame could bend the rules of distributed source coding, it remains non complex compared to intra coding.

4.3.2.2 Hash compression

Once the hash information cleverly selected, the blocks are compressed and transmitted. Aaron [Aaron *et al.*, 2004a] describes very briefly how they compress the blocks: they are coarsely subsampled and quantized (in the pixel domain), and for blocks where no hash is transmitted, a specific codeword is sent.
Ascenso [Ascenso, Pereira, 2007] proposed to compress the blocks in the DCT domain. The subbands are quantized and not all of them are transmitted. More precisely, the encoder has a fixed maximum energy, δ, to transmit and selects the n first bands (in the zigzag order) where n is the maximum number of bands such as the total energy is lower than δ. The number n is fixed for each frame. Then, the decoder makes the difference between the obtained hash code and the previous hash blocks, in order to reduce the dynamic range of the coefficients. At the end, the obtained stream is entropy coded. Moreover, the encoder builds a binary image which indicates if the hash information is transmitted or not. This map is also compressed and sent to the decoder.
Yaacoub *et al.*. [Yaacoub *et al.*, 2009a; Yaacoub *et al.*, 2009b; Yaacoub *et al.*, 2009c] also work in the DCT domain, and transmit (1/8) of the DCT coefficients.

4.3.3 Hash based side information generation methods

4.3.3.1 Hash motion estimation / interpolation

The hash information for a block at the decoder can be seen as a coarse version of the original frame. More precisely, if $\mathbf{b}$ is a block, the compression of this block can be seen as an irreversible transformation t. The generated hash for this block is thus $t(\mathbf{b})$. The purpose for the decoder is to find in the reference frames a block, $\mathbf{b}'$ whose transformation, $t(\mathbf{b}')$, is similar to $t(\mathbf{b})$.
In [Aaron *et al.*, 2004a], the method is just a simple motion estimation with a modified criterion (the SSD between the blocks is replaced with the SSD between the transformations of these blocks). If no hash is received, the decoder takes the corresponding block in the previous frame.
In [Ascenso, Pereira, 2007] the technique is a little more developed. The hash motion estimation is bidirectional (previous and next key frames) and then uses past and future information which enhances the motion search precision. Moreover, when the side information is built (based on hash, previous and next frame) the side information and the received hash are merged in a multiplexer.
Tagliasacchi *et al.*. in [Tagliasacchi, Tubaro, 2007] also perform hash based motion estimation and propose a rate distortion analysis of the hash based scheme.

4.3.3.2 Genetic algorithm fusion

Based on the principles of evolution and natural genetics, Genetic Algorithms (GAs) [Goldberg, 1989] are well suited for non-linear optimization problems. Yaacoub *et al.*. [Yaacoub *et al.*, 2009a; Yaacoub *et al.*, 2009b; Yaacoub *et al.*, 2009c] use a GA in a fusion-based approach and aim at improving the quality of the side information relying on several initial estimations.

This algorithm was integrated in one of our contribution, we thus give its detailed description in the corresponding chapter (in Section 7.1.3). In a nutshell, the genetic fusion algorithm principle is to merge different estimations by using the evolution and natural genetic laws. The results obtained by Yaacoub *et al.*. are convincing concerning the benefits of using this kind of approach.

4.4 Conclusion

We have seen that the problem of side information generation is really popular (a high number of existing methods) but also very complex (a great specificity of each problem). Whereas the developped techniques improve the SI quality, they are designed for very particular conditions and become inefficient as soon as the configuration is slightly modified. That is especially the case of the fusions methods which differs from the available frames, and from the quality of the estimation to merge. In the next chapters, we propose several techniques in order to enhance the side information generation process for several configurations (temporal and inter-view interpolations, fusions and hash-based schemes).

Chapter 5

ESSOR project scheme

The ESSOR *project (codagE de SourceS vidéO distRibué), funded by French ANR, gathered several research departments (IRISA Rennes, LSS Supélec, I3S Nice, LTCI* TÉLÉCOM *ParisTech) with the target of investigating several aspects of distributed source coding. For monoview distributed video coding we developed a new wavelet-based scheme, with a novel interpolation method. In Section 5.1, we explain the general structure of the proposed scheme and we detail some parts of it. In Section 5.2 we detail the side information generation technique, and finally in Section 5.3 we illustrate with some experimental results.*

Contents

5.1 A wavelet based distributed video coding scheme 128
5.1.1 Key Frame Encoding and Decoding 128
5.1.2 Wyner Ziv Frame Encoding 129
5.1.3 Wyner-Ziv Frame Decoding 132
5.2 Proposed interpolation method 134
5.3 Experimental results 136
5.3.1 Lossless Key frames 136
5.3.2 Lossy Key frame encoding with H.264 Intra 136
5.3.3 Lossy Key frame encoding with JPEG-2000 137
5.3.4 Interpolation error analysis 138
5.3.5 Rate-distortion performances 138
5.4 Conclusion 139

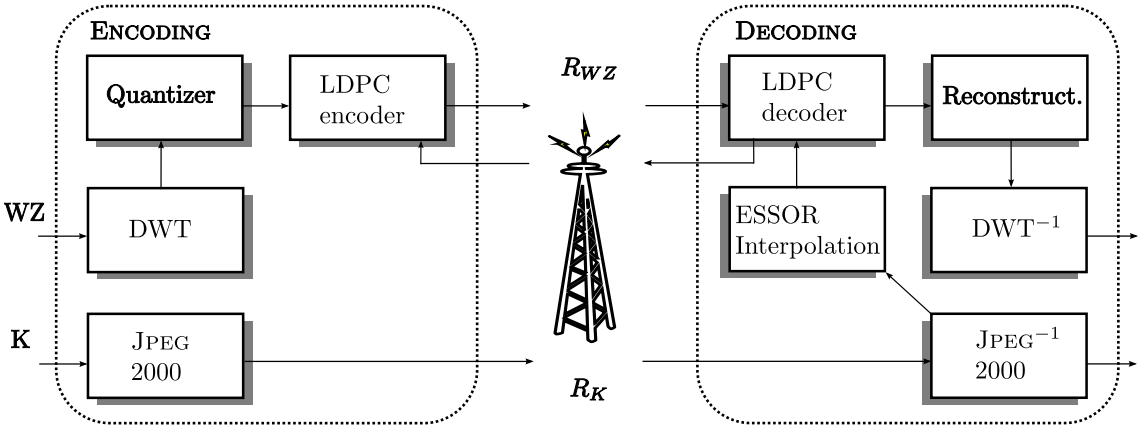


Figure 5.1: Wavelet based distributed video coding scheme adopted by the ESSOR project.

5.1 A wavelet based distributed video coding scheme

The ESSOR codec architecture is inspired from the Stanford approach just as the DISCOVER scheme [DISCOVER-website, 2005]. The differences with DISCOVER are twofolds: firstly both the intra and WZ coding use wavelets (instead of DCT), and secondly, the interpolation algorithm is different (see Section 5.3 for more details). However, the functional blocks of the ESSOR Codec (Figure 5.1) follow the same principles as all Stanford-based schemes:

1. *Partition of the GOP:* The way of partioning the K and WZ frames within a GOP (similar to DISCOVER, not detailed here).

2. *K frame coding:* Encoding and decoding of K frames with a still image codec.

3. *WZ encoding:* Encoding of a WZ frame, including the DWT, quantification of coefficients, and accumulate LDPC coding of each bitplane.

4. *SI construction:* Construction of SI using K frames or/and the previously constructed WZ frames.

5. *WZ decoding:* Decoding of a WZ frame using reconstructed SI frame and the syndrome bits of the WZ frame. This process covers the residual error estimation, LDPC decoding, and reconstruction of the WZ frame.

Following sections describe the details of the main blocks of the scheme.

5.1.1 Key Frame Encoding and Decoding

The KFs are separately encoded by the still image compression standard JPEG2000. The reference implementation, Verification Model (VM) 8.5, is adopted. The JPEG2000 encoder is presented in Figure 5.2. The key element of the JPEG2000 encoder is the EBCOT algorithm (Embedded Block Coding with Optimized Truncation) [Taubman, 2000], which can be divided into two parts. In the first part, each quantized subband is divided into blocks, called code-blocks. These code-blocks are independently coded, with a bitplane

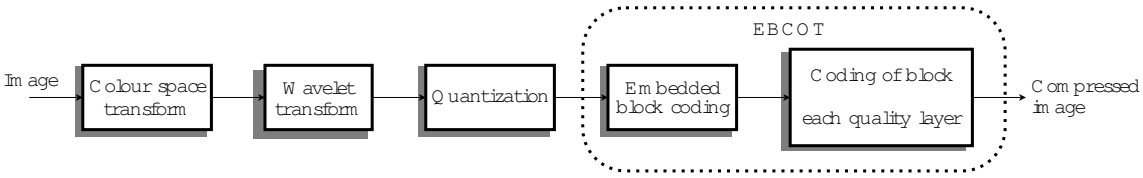


Figure 5.2: The JPEG2000 encoder.

arithmetic encoder. A rate-distortion curve is computed for each code-block and is used by the second part of EBCOT to create the final bitstream by allocating to each code-block a bit budget such that the total distance is minimized given the available bitrate. This stream, composed of EBCOT packets organized in quality layers, can be reordered depending on the desired scalability. The main features of JPEG2000 come from the use of a wavelet transform (resolution scalability), a bitplane-by-bitplane coding (quality scalability), a code-block coding (spatial random access) and a flexible organization of the codestream (manipulations in the compressed domain).

5.1.2 Wyner Ziv Frame Encoding

In Essor architecture, WZ frames are encoded in three steps. First of all the DWT is applied to the frame, then each coefficient is uniformly quantized using one of the predefined step sizes. Finally, each bitplane of each quantized frequency subband is coded with accumulate LDPC code. The details of the each step can be found in the following sections.

5.1.2.1 Discrete Wavelet Transform and quantization

A separable transform is used for the WZFs in order to perform the dyadic decomposition of an entire frame into frequency subbands (see Figure 5.3). For each frame, the rows and columns are successively decomposed over two levels of decomposition of a DWT using a fast lifting implementation of the discrete biorthogonal CDF 9/7 filter (proposed by Cohen, Daubechies and Feauveau in [Cohen *et al.*, 1992]), which results into one LL subband (horizontal and vertical low frequencies), two LH subbands (horizontal low frequencies and vertical high frequencies), two HL subbands (horizontal high frequencies and vertical low frequencies) and two HH subbands (horizontal and vertical high frequencies) as shown in Figure 5.3. It is used as the default filter in the irreversible wavelet transform of JPEG2000 due to its good compression performance. The pair defines 9 coefficients for the lowpass filter and 7 coefficients for the highpass filter of the analysis decomposition, all having irrational coefficients. The wavelet decomposition is performed only on two levels, because with more levels, the number of coefficient in the LL band would become too small and this would affect the LDPC efficiency (not adapted for too small bitstreams).

A uniform scalar quantization is then used for the approximation subband. For the detail subbands, a dead-zone quantization is applied. The Essor codec uses 8 different Quantization Indexes (QI) in order to adjust the rate allocation of the WZ frames. Each QI is associated to a set of quantization steps (one per band). The number of quantization levels for each subband in this setting are shown in Table 5.1. The quantized coefficients of each subbands are then organized in bitplanes and given as input to the channel encoder.

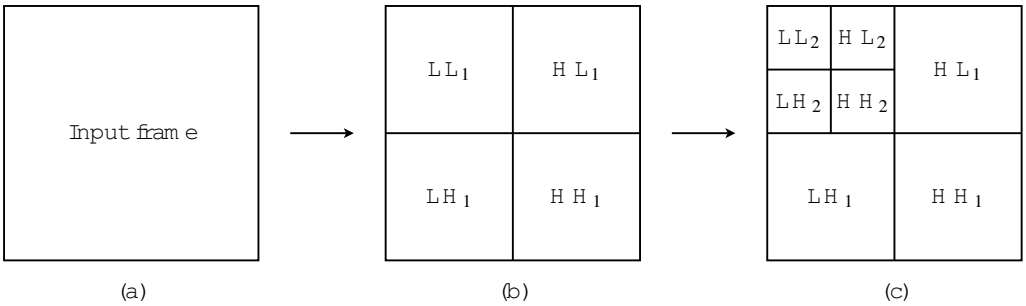


Figure 5.3: Dyadic decomposition of an input frame (a) in frequency subbands after one (b) and two (c) decomposition levels.

Table 5.1: 8 quantization indexes used for controling the WZ quantization precision. The 8 tables indicate the number of levels used to describe each band.

QI=1				QI=2				QI=3				QI=4			
16	0	0	0	16	8	0	0	32	8	0	0	64	16	0	0
0	0	0	0	8	0	0	0	8	8	0	0	16	16	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QI=5				QI=6				QI=7				QI=8			
64	32	0	0	64	32	4	4	64	32	16	16	128	64	32	32
32	16	0	0	32	32	4	4	32	32	16	16	64	64	32	32
0	0	0	0	4	4	0	0	16	16	8	8	32	32	16	16
0	0	0	0	4	4	0	0	16	16	8	8	32	32	16	16

5.1.2.2 Accumulate LDPC coding

Low Density Parity Check (LDPC) codes have been first proposed by [Gallager, 1963] and reinvented by [Mackay, Neal, 1997]. A k/n rate linear binary (n, k) LDPC Code is a block code that is defined by an $(n-k) \times n$ sparse parity check matrix $\mathbf{H}$, which has only a few number of 1s in each row and column (for instance, Equation (5.1)).

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{5.1}$$

An ensemble of the LDPC codes is described by the degree distribution polynomials $\lambda(x)$ and $\rho(x)$ [Richardson *et al.*, 2001]. $\lambda(x)$ is given as

$$\lambda(x) = \sum_i \lambda_i x^{i-1}, \tag{5.2}$$

and $\rho(x)$ is defined as

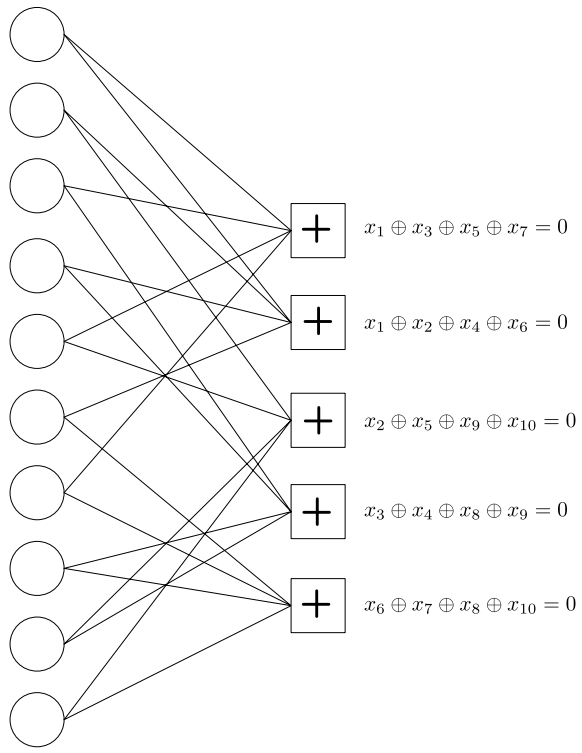


Figure 5.4: Bipartite graph representation of the parity check matrix $\mathbf{H}$.

$$\rho(x) = \sum_j \rho_j x^{j-1}, \tag{5.3}$$

where λ_i is the fraction of edges that are incident on degree-i bit nodes and ρ_j is the fraction of edges that are incident on degree-j check nodes. The rate of the LDPC code for a given pair $(\rho(x), \lambda(x))$ is bounded by

$$R \geq 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}, \tag{5.4}$$

with equality if and only if the rows of the parity check matrix are linearly independent.

The transmitter sends the syndrome $\mathbf{s} = \mathbf{H}^t\mathbf{x}$. The receiver receives the vector $\mathbf{y}$ with a transition probability $p(y|x)$. The aim of the decoder is to find the maximum likelihood codeword $\mathbf{x}_{ML} = \arg\max_x p(y|x)$. If $\mathbf{H}$ does not include cycles, the sum product algorithm converges to the exact solution [Pearl, 1988].

In our scheme, the quantized DWT coefficients of the WZFs are encoded bitplane per bitplane (from the most significant to the least significant bit) using a Slepian-Wolf encoder based on LDPC accumulate (LDPCA) codes, and only the produced accumulated syndromes are put into a buffer and sent to the decoder. LDPCA codes were described in [Varodayan *et al.*, 2005] as an efficient way of using LDPC codes in a rate-adaptive distributed source coding scheme. The LDPCA encoder consists of an LDPC syndrome-former concatenated with an accumulator (see Figure 5.5). The LDPCA decoder changes its decoding graph each time it receives an additional increment of accumulated syndromes.

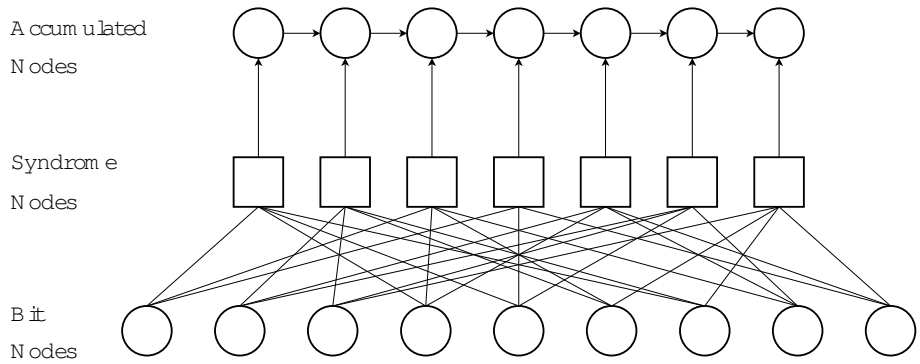


Figure 5.5: The LDPCA encoder.

This structure enables a *smooth* rate-adaptivity where the modification of the decoding graph always maintains the degree of all source nodes. At the decoder, the SI generated from the key frames is used to decode the WZFs. The accumulated syndrome bits stored in the buffers are transmitted in small amounts upon the decoder request via the feedback channel.

5.1.3 Wyner-Ziv Frame Decoding

Wyner Ziv decoding of the Essor codec is composed of the residual error estimation, the decoding of accumulate LDPC bits, and finally an Inverse DWT is applied to the decoded wavelet coefficients. The correlation noise estimation is performed as in Discover (more details in Chapter 8).

5.1.3.1 Accumulate LDPC Decoding

The Essor codec uses the accumulated syndrome bits stored in the buffers that are transmitted gradually depending on the correct decoding. In syndrome decoding, the belief propagation algorithm is used. It can be resumed as follows.

♦ **Definitions**

- The set of bits n that participates in the check m is $\mathcal{N}(m) \equiv \{n : \mathbf{H}_{mn} = 1\}$. For example, $\mathcal{N}(1) \equiv \{1, 3, 5, 7\}$ in Figure 5.4.
- The set of checks in which bit n participates is $\mathcal{M}(n) \equiv \{m : \mathbf{H}_{mn} = 1\}$. For example, $\mathcal{M}(1) \equiv \{1, 2\}$ in Figure 5.4.
- $\mathcal{N}(m)\backslash n$ is the set $\mathcal{N}(m)$ with bit n excluded.
- q^x_{mn} is the probability of the n'th bit of vector $\mathbf{x}$, where x gives the informations obtained via checks other than check m.
- r^x_{mn} is the probability of check m satisfied if bit n of $\mathbf{x}$ is considered fixed at x and the other bits $q_{mn'} : n' \in \mathcal{N}(m)\backslash n$.
- δq_{mn} is difference between the probabilities n'th bit of the vector $\mathbf{x}$ is 0 and 1 given the informations obtained via checks other than check m, $\delta q_{mn} = q^0_{mn} - q^1_{mn}$.

- δr_{mn} is the probability check m satisfied if bit n $\mathbf{x}$ is 0 given the informations obtained via checks other than check m minus that of bit n $\mathbf{x}$ is 1, $\delta r_{mn} = r^0_{mn} - r^1_{mn}$.

♦ Initialization

Depending on the vector $\mathbf{y}$ received from the channel and the channel model, the likelihood probability $p(x_n|\mathbf{y})$ for each bit n is calculated. For instance, for a memoryless binary symmetric channel with crossover probability ρ, $p(x_1 = 0|y_1 = 0) = (1 - \rho)$ and $p(x_1 = 1|y_1 = 0) = \rho$.

q^0_{mn} and q^1_{mn} values are initialized with the corresponding likelihood probabilities received from the channel respectively, such that $q^0_{mn} = p(x_n = 0|\mathbf{y})$ and $q^1_{mn} = p(x_n = 1|\mathbf{y})$. Then each variable node sends the messages δq_{mn} to its connected check.

♦ Check node iteration

Each check node sends a message to the connecting bit j, r^a_{ij}. This message is an approximation of the probability that check i is satisfied given the symbol j is a:

$$r^a_{ij} = Pr\{\text{check i satisfied}|x_j = a\}, \tag{5.5}$$

$$r^0_{mn} \approx \sum_{x_{n'}:n'\in\mathcal{N}(m)\backslash n} p(\sum_{x_z:z\in\mathcal{N}(m)} x_z = 0\text{mod } 2|x_n = 0) \prod_{\mathcal{N}(m)\backslash n} q^{x_{n'}}_{mn'} \tag{5.6}$$

Then there is a shortcut for calculating r^a_{ij} by first calculating δr_{mn}:

$$\delta r_{mn} = \prod_{n'\in\mathcal{N}(m)\backslash n} \delta q_{mn'}, \tag{5.7}$$

where $r^0_{mn} = 1/2(1 + \delta r_{mn})$ and $r^1_{mn} = 1/2(1 - \delta r_{mn})$. The δr_{mn} can be calculated efficiently by using a backward-forward algorithm [Bahl *et al.*, 1974].

♦ Variable node iteration

In this step, the q^0_{mn} and q^1_{mn} values are calculated by using the output from the check node iteration.

$$q^0_{mn} = \alpha_{mn} p(x_n = 0|\mathbf{y}) \prod_{m'\in\mathcal{M}(n)\backslash m} r^0_{m'n}, \tag{5.8}$$

and

$$q^1_{mn} = \alpha_{mn} p(x_n = 1|\mathbf{y}) \prod_{m'\in\mathcal{M}(n)\backslash m} r^1_{m'n}, \tag{5.9}$$

where α_{mn} is a normalization factor such that $q^0_{mn} + q^1_{mn} = 1$.

♦ Final Guess

Posterior probabilities of each bit can be calculated as

$$q^0_n = \alpha_n p(x_n = 0|\mathbf{y}) \prod_{m\in\mathcal{M}(n)} r^0_{mn}, \tag{5.10}$$

and

$$q_n^1 = \alpha_n p(x_n = 1|\mathbf{y}) \prod_{m \in \mathcal{M}(n)} r_{mn}^1. \tag{5.11}$$

The estimate $\hat{x}$ can be found by just thresholding the posterior probabilities

$$\hat{x}_n = \arg\max_i q_n^i. \tag{5.12}$$

For the codeword decoding point, $\hat{x}_n$, we can check if all the check nodes are satisfying $\mathbf{H}\hat{x} = 0 \bmod 2$. If it is not the case, the check-node and variable-node iterations will be repeated respectively. The iterations halt either if a codeword is found or a maximum number of iterations is reached.

5.2 Proposed interpolation method

The material in this section was published in:

- C. Dikici, T. Maugey, M. Agostini, and O. Crave, "Efficient frame interpolation for wyner-ziv video coding," in *Proc. SPIE Visual Commun. and Image Processing*, San Jose, CA, USA, Jan. 2009.

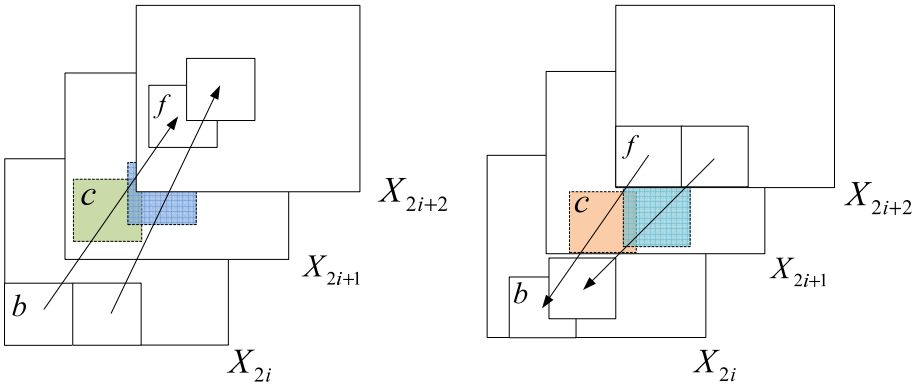
5.2.0.2 Forward and Backward motion estimation

A block matching algorithm can be used to find the best block match of the target block b in KF X_{2i} in the next KF, $X_{2(i+1)}$. The parameters that characterize the estimation technique are the block size, the matching criterion, the search range and the precision. Given that the best matching for the block b of X_{2i} in $X_{2(i+1)}$ is f with a motion vector $\vec{w_f}$, the projection of these two blocks onto the frame X_{2i+1} is $c = \frac{b+f}{2}$, where c is centred at the center of the block $b + \vec{w_f}/2$. An illustration of the forward motion estimation between X_{2i} and $X_{2(i+1)}$ and their projection on X_{2i+1} can be found in Figure 5.6(a). When the forward motion vectors are projected on the frame X_{2i+1} under the assumption of linear velocity of the motion vectors, overlapping and uncovered areas will appear. The overlapping areas correspond to the multiple motion vectors which pass through a unique pixel, whereas uncovered areas correspond to the absence of the motion trajectory through these pixels.

A similar calculation is done for the backward motion estimation (see Figure 5.6(b)), where the aim is to find the block b in X_{2i} which is the best estimation of block f in $X_{2(i+1)}$. Given a motion vector $\vec{w_b}$, the candidate block c of X_{2i+1} can be calculated similarly as in the forward case $c = \frac{b+f}{2}$, where here c is centred at $f + \vec{w_b}/2$.

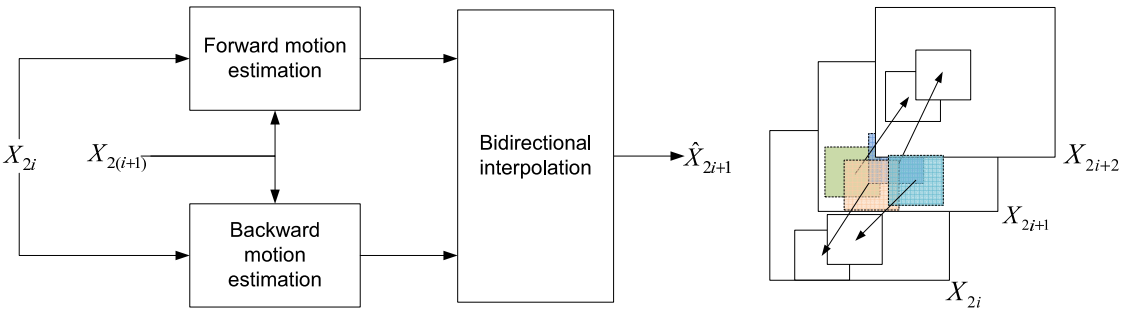
5.2.0.3 Bidirectional Interpolation

Forward and backward motion vectors ($\vec{w_f}$,$\vec{w_b}$) are computed between two key frames as explained in the previous section. We assume that there exists a linear motion between the key frames and the interpolated frames. Hence $\vec{w_f}/2$ and $\vec{w_b}/2$ are used for the motion compensation. After the forward and the backward motion compensation, a bidirectional frame interpolation step is applied as follows:



(a) Forward interpolation. (b) Backward interpolation.

Figure 5.6: Classical interpolation tools.



(a) Block diagram of KF Interpolation. (b) Forward-Backward interpolation.

Figure 5.7: ESSOR frame interpolation.

Let $p_i(x, y)$ be the pixel value of the i-th frame at the coordinates of x and y. We define the set $\mathcal{C}$ of motion compensated blocks that pass through the pixel $p_{2i+1}(x, y)$ as $\mathcal{C}(p_{2i+1}(x, y))$. Then the interpolated pixel value yields:

$$\hat{p}_{2i+1}(x, y) = \begin{cases} \frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} c_i & \text{if} \quad |\mathcal{C}| > 0, \\ 0.5 \times (p_{2i}(x, y) + p_{2i+2}(x, y)) & \text{else,} \end{cases} \tag{5.13}$$

where $|\mathcal{C}|$ is the number of members in the set $\mathcal{C}$. Hence if the set $\mathcal{C}$ is not empty, (*i.e.*,at least one motion vector passes through the pixel value $p_{2i+1}(x, y)$), then an averaging of the corresponding pixel values in the motion compensated blocks of the set $\mathcal{C}$ is performed. Otherwise, we apply a simple averaging of the pixel values in previous and next KFs. The block diagram of the ESSOR's interpolation method and the visualization of the bidirectional estimation is found in Figure 5.7. Contrary to the non overlapped block matching approach in [Ascenso *et al.*, 2005a], ESSOR's KF interpolation method allows overlapped block matching and a pixel by pixel estimation is done in the final step.

5.3 Experimental results

In order to evaluate the proposed interpolation method, we use QCIF resolution sequences with 15 fps such as *foreman*, *news* and *hall monitor* for the first 75 frames. Even frames are selected as KFs and their lossy version is available at the decoder, and the odd frames are interpolated from the KFs. We compare our results with average frame interpolation and with the methods proposed in [Ascenso *et al.*, 2005a; Ascenso *et al.*, 2006] available online at [DISCOVER-website, 2005]. In all our experiments, we use a fixed block size of 8×8 pixels, a search range of ± 16, a step size of 4 pixels for the overlapped blocks, and an integer pixel precision for the forward and the backward motion estimation. The step size determines the shift of the blocks for calculating the next motion vector, hence MV's are calculated for the overlapped blocks for every 4 pixels in height and width. We use three different KF types: lossless coding of KFs, H.264 intra-coding of KFs with different visual qualities, and JPEG-2000 coding of KFs with different visual qualities.

5.3.1 Lossless Key frames

In this section, the side information is generated using non-degraded reference frames. We compare the proposed method (Essor) to the Discover approach [Artigas *et al.*, 2007a] and the basic interpolation method (average of the two reference frames, denoted by Avg). Experimental results are presented in Table 5.2. One can see that our approach outperforms the Discover solution by up to 1.04 dB.

Table 5.2: Performance of frame interpolation methods in PSNR [dB] for lossless Key Frames.

Sequence	Avg	[Ascenso *et al.*, 2005a]	[Ascenso *et al.*, 2006]	Our method
news	39.76	39.80	39.83	**40.27**
foreman	27.86	29.42	29.79	**29.90**
hall monitor	37.84	38.57	38.69	**39.73**

5.3.2 Lossy Key frame encoding with H.264 Intra

In practical video coding contexts, the KFs are compressed, and the available KFs are not the original one anymore. In many coding schemes in the literature [Artigas *et al.*, 2007a], the coder used to encode the KFs is H.264 Intra [Wiegand *et al.*, 2003]. In this section, the proposed interpolation is compared to the Discover one, in case of H.264 Intra transmission of the KFs. We use three different quantization levels corresponding to low, medium, and high bitrates (QPs respectively equal to 40, 34 and 27). The experimental results are presented for the three test sequences in respectively Tables 5.3, 5.4, and 5.5. The respective KFs average PSNR values are given in the first row of each table. For each quantization step, we compare the average PSNR values obtained with our approach with the ones obtained by Discover and by the average method. The results show an improvement of the performance in average PSNR value compared to the Discover approach, of 0.2 dB for *news*, 0.1 dB for *foreman*, and 0.5 dB for *hall monitor*. We note

that, for low PSNR values of the KF coding, the interpolation methods can slightly surpass the average PSNR value of the KFs because the motion activity is really low.

Table 5.3: Performance of *news* sequence when KFs are coded as H-264 Intra frames with mean PSNR values 29.3 dB, 34.34 dB, and 40.7 dB.

Average KF Distortion	29.3 dB	34.34 dB	40.7 dB
Averaging	29.614	33.47	37.64
DISCOVER	29.616	33.49	37.72
ESSOR	**29.704**	**33.64**	**37.96**

Table 5.4: Performance of *foreman* sequence when KFs are coded as H-264 Intra frames with mean PSNR values 29.5 dB, 33.6 dB, and 39.9 dB.

Average KF Distortion	29.5 dB	33.6 dB	39.9 dB
Averaging	26.43	27.28	27.74
DISCOVER	27.43	28.76	29.64
ESSOR	**27.57**	**28.87**	**29.66**

Table 5.5: Performance of *hall monitor* sequence when KFs are coded as H-264 Intra frames with mean PSNR values 30.9 dB, 34.3 dB, and 40 dB.

Average KF Distortion	30.9 dB	34.3 dB	40 dB
Averaging	29.9	33.31	36.53
DISCOVER	30.05	33.73	37.30
ESSOR	**30.27**	**34.10**	**38.02**

5.3.3 Lossy Key frame encoding with JPEG-2000

While the DISCOVER approach consists in using a discrete cosinus transform (DCT) based method, in the ESSOR project, the adopted DVC scheme is based on the discrete wavelet transform (DWT). Indeed, the intra coder is chosen to transmit the KFs is JPEG-2000 [JPEG-2000, 2000]. This section provides the results obtained by this setup, and a comparison is given with the existing methods. Similar to the previous section, we produced three different levels of quantization, for the three sequences, which can be seen respectively in Tables 5.6, 5.7, and 5.8. One can see that the results of the proposed approach surpass the ones of the two other tested approaches by up to 0.9 dB in some cases.

Table 5.6: Performance of *news* sequence when KFs are Intra coded as JPEG-2000.

Average KF Distortion	29.5 dB	37 dB	41.5 dB
Averaging	29.48	35.71	38.01
Discover	29.49	35.74	38.04
Essor	**29.59**	**35.99**	**38.40**

Table 5.7: Performance of *foreman* sequence when KFs are Intra coded as JPEG-2000.

Average KF Distortion	31 dB	35 dB	41 dB
Averaging	26.97	27.70	27.8
Discover	28.11	29.40	29.73
Essor	**28.26**	**29.57**	**29.79**

Table 5.8: Performance of *hall monitor* sequence when KFs are Intra coded as JPEG-2000.

Average KF Distortion	30.9 dB	39 dB	43.4 dB
Averaging	30.53	35.78	37.17
Discover	30.72	36.43	37.94
Essor	**30.93**	**37.13**	**38.88**

5.3.4 Interpolation error analysis

As presented in the previous section, Essor interpolation method outperforms the Discover techniques. In this section we propose to analyze the behaviour of the SI error for the different methods.

Figure 5.8 represents the evolution of the PSNR of the side information along the time for QCIF *foreman* test sequence. These plots show that when the motion activity is not important, Essor method outperforms the others. This can be explained by the fact that this technique presents a smoothing property. In case of high motion activity, Discover builds an SI of higher quality than Essor.

In Figure 5.9, zooms on the side informations for the third frame of *news* test sequence are represented. Error images are also shown. Looking at these figures, one can clearly see the smoothed aspect of Essor estimation, while the SI of Discover presents some blocking artifacts.

5.3.5 Rate-distortion performances

The different blocks presented in this chapter have been implemented by us and several research members of Essor project. A complete scheme is now available, and in this section we present the rate-distortion curves obtained for several video sequences. However, the presented performances are the very first results obtained with the implemented schemes,

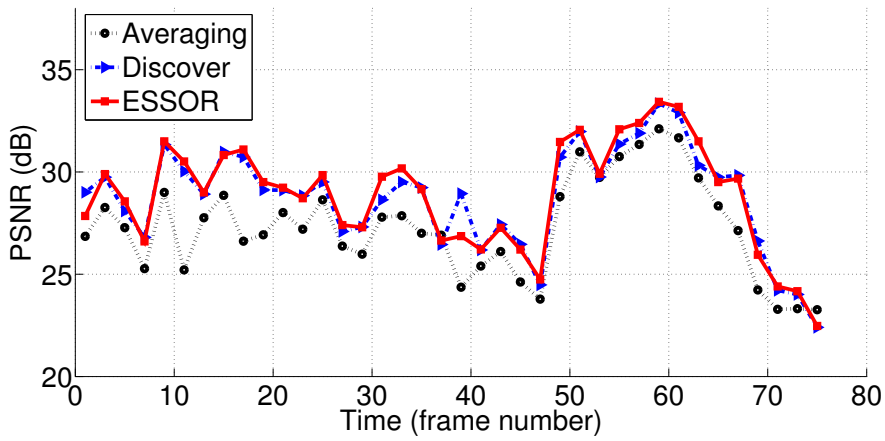


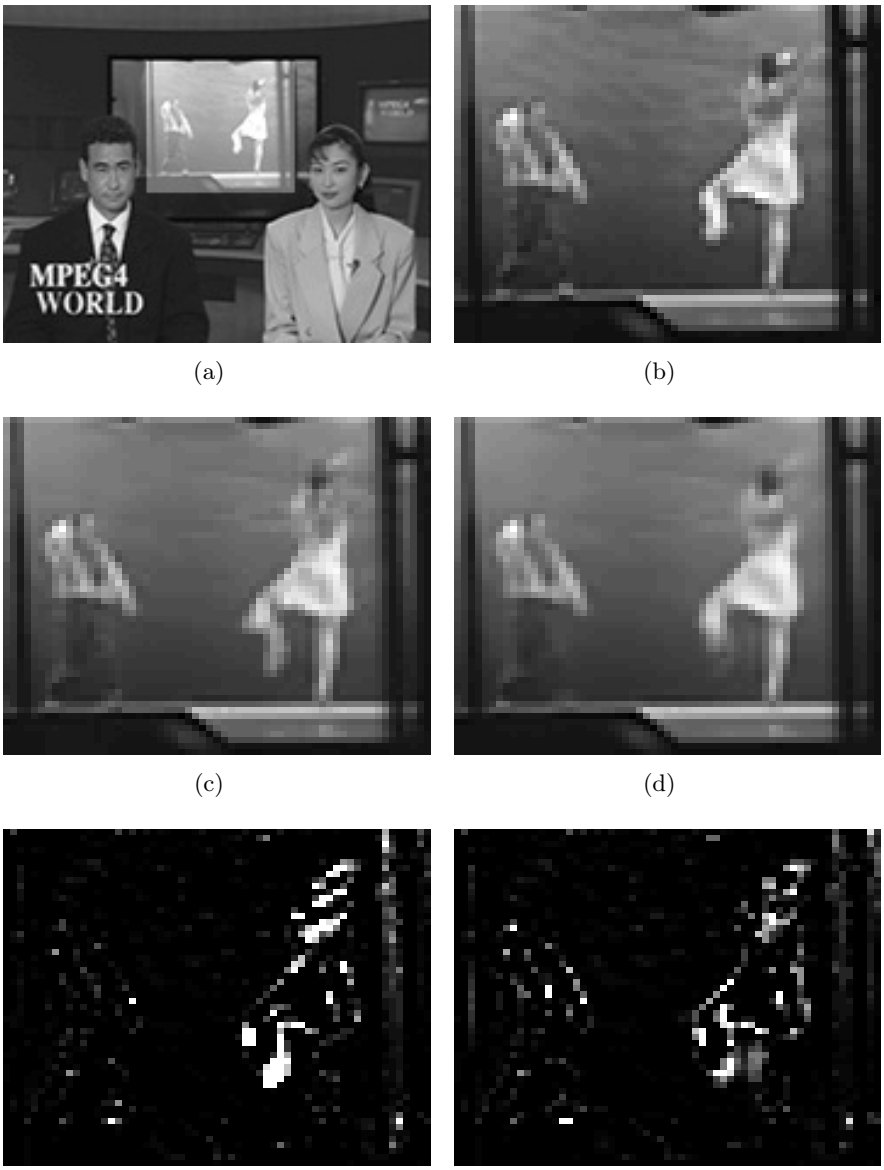
Figure 5.8: PSNR [dB] quality of each interpolated SI frame of *foreman* sequence for the three interpolation methods. K frames are quantized with JPEG2000.

and thus are not yet optimized. Indeed, several parameters have to be tested, such as the quantization matrix, the alpha calculation, the correspondence between the key frame quantization and the WZ quantization index.

Figure 5.10 displays the ESSOR decoding performance for three QCIF sequences, compared with the JPEG2000 intra coding results. For the three sequences, the ESSOR codec is more efficient than JPEG2000 intra coding.

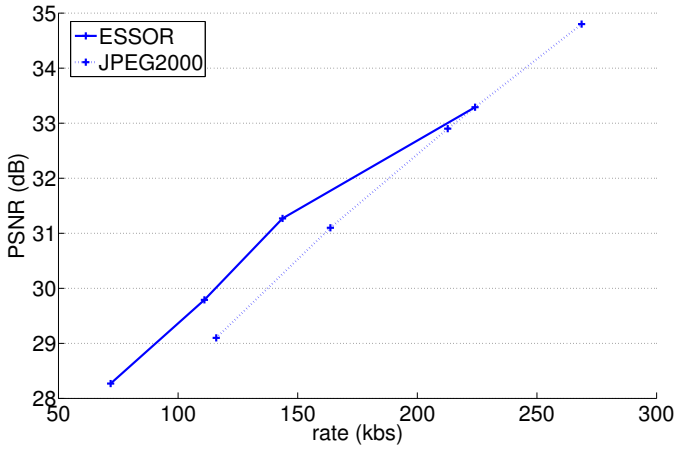
5.4 Conclusion

The proposed interpolation technique seems to be efficient and outperforms the reference for several test sequences. This algorithm has been integrated in an original coding scheme developped within the french ANR project ESSOR. Even if the results seem to be promising, they need to be further tested, optimized and finally compared to the state-of-the-art scheme DISCOVER.

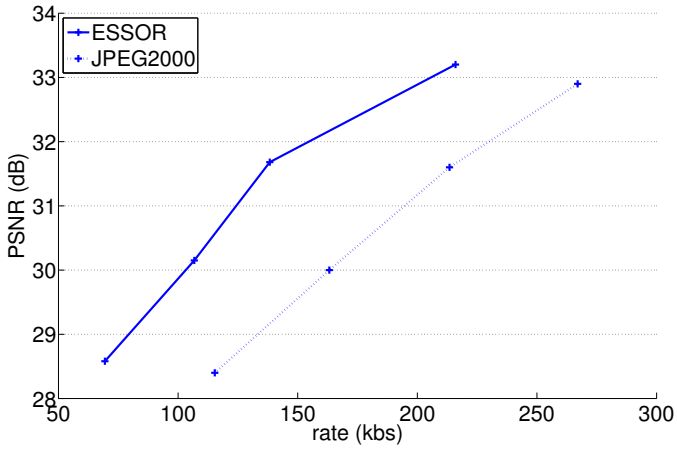


(a) (b) (c) (d) (e) (f)

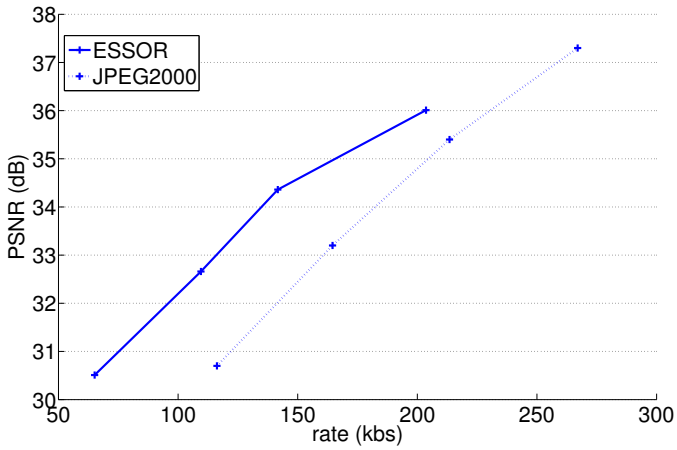
Figure 5.9: Interpolation performance of the *news* sequence, frame #3, zooming on the centre of the frame. (a)Original frame. (b)Zoom on original frame. (c)Zoom on Discover interpolation. (d)Zoom on Essor interpolation performance. (e)Zoom on Discover interpolation error. (f)Zoom on Essor interpolation error.



(a) *foreman*, QCIF, 100 frames, 15 fps



(b) *salesman*, QCIF, 100 frames, 15 fps



(c) *carphone*, QCIF, 100 frames, 15 fps

Figure 5.10: Rate-Distortion performance of ESSOR scheme compared to JPEG2000 Intra for three QCIf video sequences, 176×144.

Chapter 6

Side information refinement

Almost all of the side information generation methods developed for DVC adopt a block-based approach. This is mainly explained by two reasons. Firstly, the existing methods involve different techniques (as motion search, block vectors filtering, etc.) which were initially built for classical video coding, where the number of vectors need to be limited because of their transmission cost. This motivation is not relevant in DVC because the SI generation is performed at the decoder, and thus, the vectors are not transmitted. Therefore, the SI generation algorithms can perform their estimation pixel by pixel, which would avoid the blocking artifacts.
The second reason was given by some works which studied pixel-based motion interpolation for classical video coding [Tang, Au, 1998]. They indeed found that sometimes a pixel-based interpolation would sensibly improve the performances of a block-based motion interpolation by avoiding the blocking artifacts, but on the other hand, pixel-based methods can sometimes degrade the quality of the estimation by adding a salt-and-pepper effect. Another drawback of pixel-based approaches is their big complexity. However, this disadvantage is not seriously considered in DVC where the decoder computation capacity is assumed to be high anyway.
In this chapter, we propose to study pixel-based, dense*, SI estimation in the DVC context. Firstly, in Section 6.1, we propose a family of dense interpolation methods, which are based on two refinement techniques: the Cafforio-Rocca algorithm [Cafforio, Rocca, 1983] and a total-variation based method proposed by Miled [Miled* et al.*, 2009]. Then in Section 6.2, for a multiview context, we propose several fusion methods which aim at merging temporal and inter-view interpolations.*

Contents

6.1 Generation of dense vector fields 144
6.1.1 Motivations and general structure 144
6.1.2 Cafforio-Rocca algorithm (CRA) 145
6.1.3 Total variation based algorithm 154
6.1.4 Experiments 159
6.2 Proposed fusion methods 164
6.2.1 Recall of the context 164
6.2.2 Proposed techniques 164
6.2.3 Experimental results 166
6.3 Conclusion 168

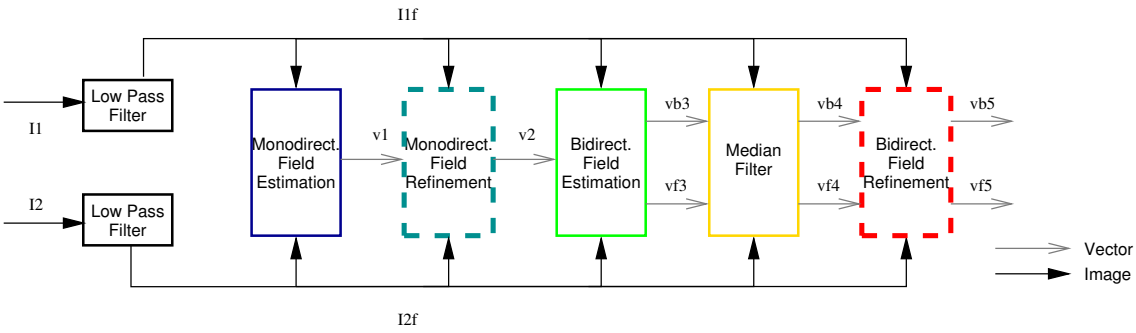


Figure 6.1: Structure of proposed interpolation scheme.

6.1 Generation of dense vector fields

6.1.1 Motivations and general structure

As explained above, we investigate here the efficiency of dense (one vector per pixel) interpolation methods for temporal and inter-view estimations. We thus propose several estimation techniques, all of them based on the Discover interpolation algorithm. Indeed, this block-based frame estimation scheme is one of the most efficient interpolation technique in the literature, and it could thus be interesting to transpose it to a pixel-based approach. However, a naive adaptation (for example a decrease of the block size to 1) would product the bad effects highlighted in [Tang, Au, 1998], *i.e.*,salt-and-pepper artefacts. That is why our technique keeps the Discover scheme and adds two refining blocks which aim at avoiding pixel estimation drawbacks by adopting a differential-based approach.

The classical Discover scheme is based on the following three main steps: monodirectional field estimation (mono-FE), bidirectional field estimation (bi-FE) and median filtering. The novelty of our approach is to introduce a first vector field refinement stage between the mono-FE and the bi-FE and a second one after the median filter, at the very end of the chain. The complete image interpolation scheme, proposed in this work, is represented in Figure 6.1. Two algorithms are proposed for both refinements: a first one inspired by Cafforio-Rocca works presented in Section 6.1.2 and another one based on total variation presented in Section 6.1.3. For each refinement block three possibilities are possible: Discover (D) with no refinement, Cafforio-Rocca (C) and total variation (V), which leads to nine possible schemes denoted by XY, where $X \in \{D, C, V\}$ corresponds to the first refinement block, and $Y \in \{D, C, V\}$ corresponds to the second refinement block. For example, the initial Discover scheme is denoted by DD, and a simple Cafforio-Rocca monodirectional refinement is denoted by CD.

In the following, I_b and I_a denote the two reference frames which have been low-pass filtered (the two reference frame non filtered, I_b^{input} and I_a^{input} correspond to the decoded key or WZ reference frames). They can belong to the same camera for motion interpolation, or they can belong to different cameras in case of inter-view estimations. Moreover, as presented in Figure 6.1, the monodirectional block based vector field is denoted by $\mathbf{u}_{ab}$. After the first block refinement, it is denoted by $\mathbf{u}^*_{ab}$. Similarly, the bidirectional vector fields are denoted by $\mathbf{u}_a$ and $\mathbf{u}_b$ before refinement and $\mathbf{u}^*_a$ and $\mathbf{u}^*_b$ after.

The next sections are the presentation of each refinement algorithm principles. Each

of them is firstly independently tested in its natural configuration, *i.e.*,the Cafforio-Rocca based algorithm are tested in a monoview scheme while total-variation based ones are tested in a stereo context. In Section 6.1.4, all the methods are tested and compared on the same database.

6.1.2 Cafforio-Rocca algorithm (CRA)

The first refinement algorithm we propose to introduce in the DISCOVER interpolation scheme is the Cafforio-Rocca (CR) technique [Cafforio, Rocca, 1983], which is one of the most popular motion estimation techniques in classical monoview video analysis. The CR ME algorithm is *pel-recursive*, meaning that the MV computed for the last pixel (or more generally, a function of the previous MVs) is used as initialization for the current pixel processing. The pixels are not necessary scanned in raster order; rather, an order that better preserves the correlation between successively processed pixels is often preferred, *e.g.* by scanning the even lines from the left to the right and the odd ones from the right to the left.

The original CRA consists in applying, for each pixel $\mathbf{p}$ of the image, three steps, until the estimated MV $\mathbf{u}(\mathbf{p})$ is obtained.

Initialization. Some *a priori* information is used as initialization value, $\mathbf{u}^{(1)}(\mathbf{p})$. Often the vector computed for the previous position is used for initialization.

Validation. The motion-compensated error $A = |I_a(\mathbf{p}) - I_b(\mathbf{p} + \mathbf{u}^{(1)})|$ is compared to the non-compensated error, incremented by a positive quantity γ: $B = |I_a(\mathbf{p}) - I_b(\mathbf{p})| + \gamma$. If $A \leq B$ the initialization vector is validated and kept for the next step: $\mathbf{u}^{(2)} = \mathbf{u}^{(1)}$. Otherwise, the null vector is used: $\mathbf{u}^{(2)} = \mathbf{0}$. The validation step allows to prevent algorithm divergence and to get rid of outliers, which can occur for example when the initialization vector belongs to a different object with respect to the current position. Of course, it may happen that the non-compensated error is smaller than the compensated error even if the current vector is not an outlier: the threshold value γ allows to control the number of validated vectors which are reset to zero.

Refinement. The last step consists in refining the validated vector $\mathbf{u}^{(2)}$ by adding to it a correction $\delta\mathbf{u}$. This correction is obtained by minimizing the energy of the prediction error, under a constraint on the norm of the correction vector. The Lagrangian cost function is then:

$$J(\delta\mathbf{u}) = [I_a(\mathbf{p}) - I_b(\mathbf{p} + \mathbf{u}^{(2)} + \delta\mathbf{u})]^2 + \lambda\|\delta\mathbf{u}\|^2 \tag{6.1}$$

Using a first order expansion of I_b, it turns out that the value of $\delta\mathbf{u}$ minimizing J is:

$$\delta\mathbf{u}(\mathbf{p}) = \frac{-\epsilon\phi}{\lambda + \|\phi\|^2} \tag{6.2}$$

where $\epsilon = I_a(\mathbf{p}) - I_b(\mathbf{p} + \mathbf{u}^{(2)})$ is the prediction error associated to the MV $\mathbf{u}^{(2)}$, and $\phi = \nabla I_b(\mathbf{p} + \mathbf{u}^{(2)})$ is the spatial gradient of the motion compensated reference image.

6.1.2.1 Monodirectional refinement

The material in this section was published in:

- M. Cagnazzo, W. Miled, T. Maugey, and B. Pesquet-Popescu, "Image interpolation with edge-preserving differential motion refinement," in *Proc. Int. Conf. on Image Processing*, Cairo, Egypt, Nov. 2009.

6.1.2.1.a Principle

Now we describe the CRA modifications needed in the context of DVC image interpolation. The three steps are modified and moreover we use a different scanning order, based on the blocks used in the forward field estimation: the blocks are scanned in a raster scan order, and the same is done for the pels within each block.

Our monodirectional version of the CRA takes as input $\mathbf{u}_{ab}$, the MVF produced by the forward ME (see Figure 6.1). These vectors are used in the initialization step: if $\mathbf{p}$ is the first position (*i.e.*, top and leftmost) in the block, the vector $\mathbf{u}^{(1)}(\mathbf{p})$ is initialized with $\mathbf{u}_{ab}(\mathbf{p})$. Otherwise, we use a weighted average of the left, up, and up-right neighboring vectors, with different weights if the neighbors are in the same block or not.

As far as the validation step is concerned, we not only compute the compensated error associated to $\mathbf{u}^{(1)}(\mathbf{p})$ ($A = |I_a(\mathbf{p}) - I_b(\mathbf{p} + \mathbf{u}^{(1)}(\mathbf{p}))|$) and the non-compensated error ($B = |I_a(\mathbf{p}) - I_b(\mathbf{p})|$), but also the compensated error associated to $\mathbf{u}_{ab}(\mathbf{p})$ ($C = |I_a(\mathbf{p}) - I_b(\mathbf{p} + \mathbf{u}_{ab}(\mathbf{p}))|$), and we choose the vector with the least absolute error. As in the original algorithm, the non-compensated error is increased by a threshold γ in order to reduce the reset frequency.

The new validation step allows us to reintroduce the $\mathbf{u}_{ab}(\mathbf{p})$ as validated vector while scanning the current block. This is useful, since, independently from the scanning order, it can happen that, within the same block, we pass several times from one object to another. At the first object boundary crossing, the MV is likely reset by the validation pass, then the pel-recursive nature of the CRA allows to reconstruct the MV of the new object by accumulating the corrections from one pel to the other. However, if during the scanning we come back to the first object, with the original CRA we can only reset to zero the MV; with this modification, we can benefit of a fast recovery of the first object MV.

In the last step, we refine the validated MV $\mathbf{u}^{(2)}(\mathbf{p})$ by adding a correction $\delta\mathbf{u}$. Like in the original algorithm, the correction should minimize the prediction error, under the constraint of a regularization condition. In the original algorithm it is possible to find a closed form of the optimal solution when the regularization is simply a constraint on the correction norm. Here we want to use a stronger constraint. Namely, we consider the diffusion matrix $\mathbf{D}(\nabla I)$:

$$\mathbf{D}(\nabla I) = \frac{1}{|\nabla I|^2 + 2\sigma^2}\left[\begin{pmatrix}\frac{\partial I}{\partial y}\\ -\frac{\partial I}{\partial x}\end{pmatrix}\begin{pmatrix}\frac{\partial I}{\partial y}\\ -\frac{\partial I}{\partial x}\end{pmatrix}^T + \sigma^2\mathbf{I}_2\right]$$

We use $\mathbf{I}_2$ to refer to the 2×2 identity matrix. When the regularization constraint takes into account the diffusion matrix, one is able to inhibit blurring of MV field across

object boundaries [Nagel, Enkelmann, 1986] [Alvarez, Sanchez, 2000]. This kind of constraint is well known in the literature of optical flow motion estimation and is called Nagel-Enkelmann constraint [Nagel, Enkelmann, 1986]. We propose therefore the following cost function:

$$J(\delta\mathbf{u}) = [I_a(\mathbf{p}) - I_b(\mathbf{p} + \mathbf{u}^{(2)} + \delta\mathbf{u})]^2 + \lambda\delta\mathbf{u}^T\mathbf{D}\delta\mathbf{u} \tag{6.3}$$

where we used the shorthand notation $\mathbf{D} = \mathbf{D}(\nabla I_b)$. We notice that, in the homogeneous regions where $\sigma^2 \gg |\nabla I_b|^2$, the cost function becomes equivalent to the one used in the original algorithm, see Equation (6.1).

Here we show that even with the new cost function, a closed form of the optimal vector refinement exists, and we give it at the end of this section. Like in the original algorithm, the first step is a first order expansion of the cost function:

$$J \approx \left[I_a(\mathbf{p}) - I_b(\mathbf{p} + \mathbf{u}^{(2)}) - \nabla I_b(\mathbf{p} + \mathbf{u}^{(2)})^T\delta\mathbf{u}\right]^2 - \lambda\delta\mathbf{u}^T\mathbf{D}\delta\mathbf{u} = \left(\epsilon + \phi^T\delta\mathbf{u}\right)^2 + \lambda\delta\mathbf{u}^T\mathbf{D}\delta\mathbf{u},$$

where we defined the compensation error $\epsilon = I_a(\mathbf{p}) - I_b(\mathbf{p} + \mathbf{u}^{(2)}(\mathbf{p}))$ and the compensated gradient $\phi = \nabla I_b(\mathbf{p} + \mathbf{u}^{(2)}(\mathbf{p}))$. Then we look for the refinement $\delta\mathbf{u}^*$ which minimizes the function cost: we set to zero the partial derivatives of J.

$$\mathbf{0} = \frac{\partial J}{\partial\delta\mathbf{u}}(\delta\mathbf{u}^*) = 2(\epsilon\phi^T\delta\mathbf{u}^*)\phi + 2\lambda\mathbf{D}\delta\mathbf{u}^* = 2\left(\phi\phi^T + \lambda\mathbf{D}\right)\delta\mathbf{u}^* + 2\epsilon\phi. \tag{6.4}$$

Note that the derivative of $\delta\mathbf{u}^T\mathbf{D}\delta\mathbf{u}$ has been computed in Equation (6.4) using the symmetry of $\mathbf{D}$. The last equation is equivalent to:

$$\delta\mathbf{u}^* = -\left(\phi\phi^T + \lambda\mathbf{D}\right)^{-1}\epsilon\phi.$$

Using the matrix inversion lemma, we find the optimal update vector:

$$\delta\mathbf{u}^* = \frac{-\epsilon\mathbf{D}^{-1}\phi}{\lambda + \phi^T\mathbf{D}^{-1}\phi}. \tag{6.5}$$

It is interesting to observe the similitude between the final formula and the original one in Equation (6.2). Actually, Equation (6.5) reduces to Equation (6.2) in homogeneous regions or for very high values of the parameter σ.

For parameter setting, we have run several experimental tests over a set of 4 test sequences, characterized by different motion content: *city*, *eric*, *foreman*, and *mobile* (352× 288, 30 fps). First, we have performed some experiments in order to tune the parameters λ, γ, and σ of the proposed algorithm. We look for the parameter values maximizing the PSNR between the reconstructed and the original WZF. We show some results for λ, in Table 6.1. We report the average PSNR of reconstructed WZF for different values of the parameters, averaged over the test sequences, and with KFs encoded at QP=31. Similar results were obtained for other quantization steps. We conclude that the best value for the parameters are $\lambda = 2000$, $\gamma = 20$ and $\sigma = 50$. These values will be used in the following.

6.1.2.1.b First experiments

In this section, we first present experimental results for the *CD* method over several monoview video sequences. This method will be further tested for other configurations

λ	500	1000	2000	3000	5000
PSNR [dB]	30.31	30.46	30.57	30.52	30.50

Table 6.1: Impact of λ parameter on side information quality in the CD method. Average over the four test sequences, QP=31.

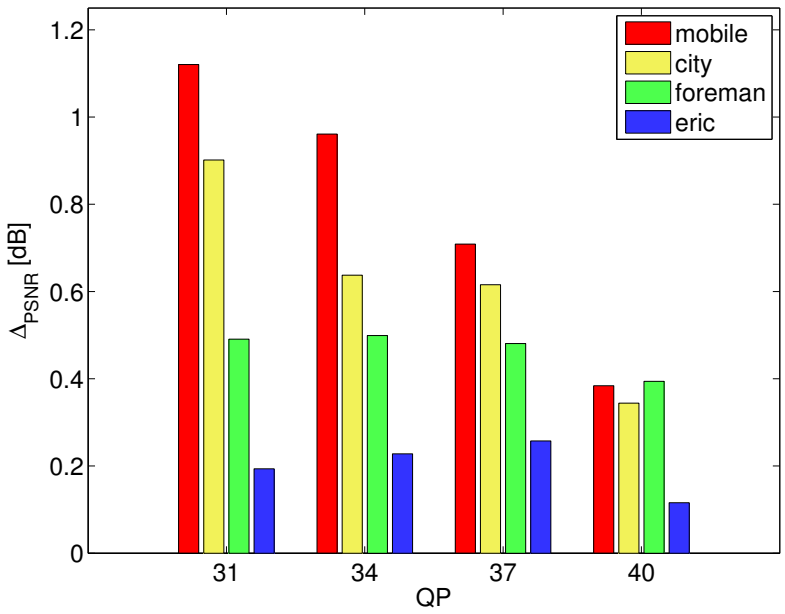


Figure 6.2: SI PSNR improvement [dB] between DD (reference) and CD.

in Section 6.1.4. We have used the same set of 4 test sequences, characterized by different motion content: *city*, *eric*, *foreman*, and *mobile*. In order to evaluate the effectiveness of the proposed technique, we first compared the SI produced by CD with the one produced by Discover (DD) using out set of four input sequences. The criterion considered for the comparison was the PSNR between the original WZF and its estimation produced by each of the techniques.

The results of the first tests are summarized in Figure 6.2. We note that for each sequence and for each KF's quantization step, CD produces a SI more similar to the original WZF (in the sense of the PSNR). However the gain can be quite different according to the sequence. We obtain higher gain when there is high, regular motion like in *mobile* and *city* (up to more than 1.1 dB). When the motion is less regular we have a bit smaller but still significant gain (up to about 0.5 dB for *foreman*). Finally, some gains are still obtained for the sequence *eric*, around 0.2 dB. We observe as well that the gain is generally smaller for severely quantized KFs: this is reasonable since low quality KFs provide a less reliable gradient information, which is at the basis of Cafforio-Rocca approach.

These first experiments were conducted for a GOP size of 2, *i.e.*, KFs are interleaved, one by one with the WZFs. We repeated the same experiment for larger GOPs, and we found that the proposed CD method is still better than the reference DD, even though the gap becomes smaller. The results of these tests are reported in Table 6.2. Even in the least favorable case of a GOP size of 8, CD is almost 0.2dB better than DD.

We then computed the global RD performance of the scheme for the sequences of the

	QP values			
GOP size	31	34	37	40
2	0.68	0.58	0.52	0.31
4	0.38	0.33	0.28	0.22
8	0.23	0.23	0.22	0.18

Table 6.2: SI PSNR improvement [dB] of CD over DD (reference) for different GOP sizes, average over the test set.

test set. The results were compared with those of the reference DISCOVER coder using the Bjontegaard metric [Bjontegaard, 2001] at four operational points corresponding to QP $\in$ $\{31, 34, 37, 40\}$. We observed an average rate reduction of 5.9% and an average PSNR improvement of 0.32 dB for the sequences of the test set. These results validate the CD proposed method.

6.1.2.2 Bidirectional refinement

The material in this section was published in:

- M. Cagnazzo, T. Maugey, and B. Pesquet-Popescu, "A differential motion estimation method for image interpolation in distributed video coding," in *Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, Taipei, Taiwan, Apr. 2009.

6.1.2.2.a Principle

We propose a new version of the CR algorithm, allowing us to obtain better ME for Wyner-Ziv frames in the context of DVC. With respect to the original algorithm, we do not dispose any more of the frame to be estimated but only of the encoded version of the adjacent KFs. We will still refer to these images as I_b and I_a. Moreover we want to exploit the block-based MVFs produced by the DISCOVER algorithm, $\mathbf{u}_a$ and $\mathbf{u}_b$.

Our ME algorithm still consists in the initialization, validation and refinement steps; but they are modified to fit the new context; moreover we use a different scanning order, based on the blocks used in the DISCOVER algorithm. A raster scan order between blocks can be used, however it is worth noting that the blocks are processed independently, so the algorithm lends itself to a parallel implementation. Within each block the positions are scanned so as to keep a high correlation between consecutively scanned positions. A possible scanning order is shown in Figure 6.3.

The initialization of the backward and forward vectors for the current position $\mathbf{p}$ is different if it is the first position (*i.e.*, top and leftmost, as highlighted in Figure 6.3) in the current block or not. In the first case, we use the MV estimated for the current block by the DISCOVER algorithm; otherwise, we recursively use the MV produced by our algorithm for the last scanned position. We call $\mathbf{u}_a^{(0)}(\mathbf{p})$ and $\mathbf{u}_b^{(0)}(\mathbf{p})$ (or *a priori*) the backward and forward vectors obtained from the initialization step.

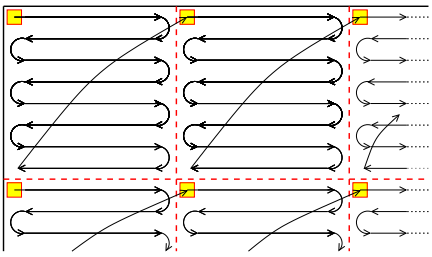


Figure 6.3: Scan order for the proposed algorithm. Highlighted position are initialized with the input MVF; others with the MV of the previously scanned position.

The validation step amounts to computing the quantities:

$$\begin{aligned} A &= |I_a(\mathbf{p}+\mathbf{u}_a^{(0)}(\mathbf{p})) - I_b(\mathbf{p}+\mathbf{u}_b^{(0)}(\mathbf{p}))| \\ B &= |I_a(\mathbf{p}) - I_b(\mathbf{p})| + \gamma, \\ C &= |I_a(\mathbf{p}+\mathbf{u}_a(\mathbf{p})) - I_b(\mathbf{p}+\mathbf{u}_b(\mathbf{p}))|. \end{aligned}$$

If A (resp. B or C) is the least quantity, we use $\mathbf{u}_a^{(0)}$ and $\mathbf{u}_b^{(0)}$ (resp. null or $\mathbf{u}_a$ and $\mathbf{u}_b$) as validated vectors. Note that, like the original CR algorithm, a threshold γ is used to penalize the reset of the estimated vector. A high threshold causes less vector resets, producing more regular but maybe less accurate motion vector fields.

In the last step, we refine the MVs at the output of the validation step, ($\mathbf{u}_a^{(1)}$ and $\mathbf{u}_b^{(1)}$) by adding a correction (d_2 and $\delta\mathbf{u}_b$). So the cost function J depends on both refinements:

$$J(d_2, \delta\mathbf{u}_b) = [I_a(\mathbf{p}+\mathbf{u}_a^{(1)}+d_2) - I_b(\mathbf{p}+\mathbf{u}_b^{(1)}+\delta\mathbf{u}_b)] + \lambda_a\|d_2\|^2 + \lambda_b\|\delta\mathbf{u}_b\|^2$$

Like in the original algorithm, the cost function is approximated by first order expansions; however here we expand both I_a and I_b:

$$\begin{aligned} J &\approx [I_a(\mathbf{p}+\mathbf{u}_a^{(1)}) + \nabla I_a(\mathbf{p}+\mathbf{u}_a^{(1)})^T d_2 - I_b(\mathbf{p}+\mathbf{u}_b^{(1)}) - \nabla I_b(\mathbf{p}+\mathbf{u}_b^{(1)})^T \delta\mathbf{u}_b]^2 + \lambda_a\|d_2\|^2 + \lambda_b\|\delta\mathbf{u}_b\|^2 \\ &= \left(\epsilon + \phi_a^T d_2 - \phi_b^T \delta\mathbf{u}_b\right)^2 + \lambda_a\|d_2\|^2 + \lambda_b\|\delta\mathbf{u}_b\|^2 \end{aligned}$$

where we defined:

$$\begin{aligned} \epsilon &= I_a(\mathbf{p}+\mathbf{u}_a^{(1)}) - I_b(\mathbf{p}+\mathbf{u}_b^{(1)}) \\ \phi_a &= \nabla I_a(\mathbf{p}+\mathbf{u}_a^{(1)}) \\ \phi_b &= \nabla I_a(\mathbf{p}+\mathbf{u}_b^{(1)}) \end{aligned}$$

Then, the actual refinements are defined as those minimizing the function cost and are found by setting to zero the partial derivatives of J. Let us start with the derivative wrt d_2.

$$\begin{aligned} \frac{\partial J}{\partial d_2} = \mathbf{0} \Leftrightarrow 2[\epsilon + \phi_a^T d_2 - \phi_b^T \delta\mathbf{u}_b]\phi_a + 2\lambda_a d_2 = \mathbf{0} \Leftrightarrow \\ [\epsilon - \phi_b^T \delta\mathbf{u}_b]\phi_a + (\lambda_a \mathbf{I}_2 + \phi_a\phi_a^T)d_2 = \mathbf{0} \Leftrightarrow d_2 = \frac{\phi_b^T \delta\mathbf{u}_b - \epsilon}{\lambda_a + \|\phi_a\|^2}\phi_a \end{aligned} \tag{6.6}$$

The last equation has been obtained by applying the matrix inversion lemma, $\left(\lambda\mathbf{I}+\mathbf{u}\mathbf{u}^T\right)^{-1} = \frac{1}{\lambda}\left(\mathbf{I}-\frac{\mathbf{u}\mathbf{u}^T}{\lambda+\|\mathbf{b}\|^2}\right)$.

Likewise, the partial derivative of J with respect to d_2 is zero iff:

$$\delta\mathbf{u}_b = \frac{\phi_a^T d_2 + \epsilon}{\lambda_b + \|\phi_b\|^2}\phi_b \tag{6.7}$$

Substituting Equation (6.7) in (6.6), and applying again the matrix inversion lemma, we can easily find the optimal refinements:

$$\delta\mathbf{u}_a^* = \frac{-\epsilon\phi_a}{\lambda_a + \|\phi_a\|^2 + \frac{\lambda_a}{\lambda_b}\|\phi_b\|^2} \tag{6.8}$$

$$\delta\mathbf{u}_b^* = \frac{\epsilon\phi_b}{\lambda_b + \|\phi_b\|^2 + \frac{\lambda_b}{\lambda_a}\|\phi_a\|^2}. \tag{6.9}$$

Since usually $\lambda_a = \lambda_b$, the previous equations further simplify into:

$$\delta\mathbf{u}_a^* = \frac{-\epsilon\phi_a}{\lambda + \|\phi_a\|^2 + \|\phi_b\|^2} \tag{6.10}$$

$$\delta\mathbf{u}_b^* = \frac{\epsilon\phi_b}{\lambda + \|\phi_a\|^2 + \|\phi_b\|^2}. \tag{6.11}$$

which are formally very similar to the original algorithm update step in Equation (6.2) but for the meaning of ϵ and the presence of the sum of the two compensated gradient norms.

In order to determine the best value for the parameters of the proposed algorithm, we have run it over four popular test sequences at CIF resolution (*eric*, *foreman*, *football* and *city*) and we have obtained the even frame interpolation. These images were compared with the original frames by computing the PSNR.

In all our experiments, the threshold γ proved to have a small influence over the global performance, given that it is greater or equal to 50, so we used this value in the following.

Then we determined the relationship between the best λ_b and λ_a. The experiments confirmed the intuition that these parameters should have very close values. For all our test sequences, and for all tested values of QP, we found that the best performance is obtained when $|\lambda_b - \lambda_a| < 0.1\lambda_a$; moreover, within this interval the performance are very consistent, with a PSNR variation of less than 0.03 dB. For the sake of brevity, we only report some of these results in Table 6.3. As a consequence, in the following we take $\lambda_b = \lambda_a$ and so we shall drop the subscript.

Finally, we looked for the best value of λ. We have computed the SI PSNR over the test sequences for several values of the parameter between 1000 and 15000. As shown in Figure 6.4 the average PSNR performance are quite consistent for $\lambda \geq 5000$, with a maximum around 7500, which has been used as value for λ in the following.

6.1.2.2.b First experiments

With the values of parameters defined in the previous subsection, we have compared the *DC* method with Discover by running them over the same test sequences and using several QPs for the KF coding. The results are summarized in Figure 6.5. We observe that the *DC* is able to improve the WZF quality, up to over 0.6 dB in the average and to over

	-1000	-500	0	500	1000
eric	32.29	32.33	32.33	32.32	32.32
football	23.19	23.19	23.19	23.17	23.17
foreman	33.86	33.89	33.90	33.89	33.89
city	27.15	27.16	27.17	27.16	27.15
Average	29.12	29.14	**29.15**	29.14	29.13

Table 6.3: PSNR of SI images over the test sequences for different $\Delta\lambda = \lambda_b - \lambda_a$ and QP=31. Average over $\lambda_a \in [1000, 10000]$.

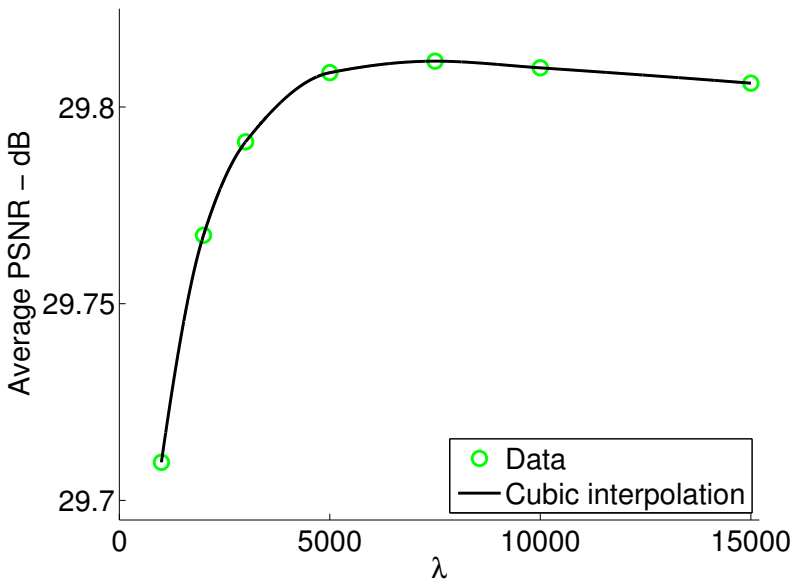


Figure 6.4: Average PSNR of side information over test sequences as a function of λ, for QP=31.

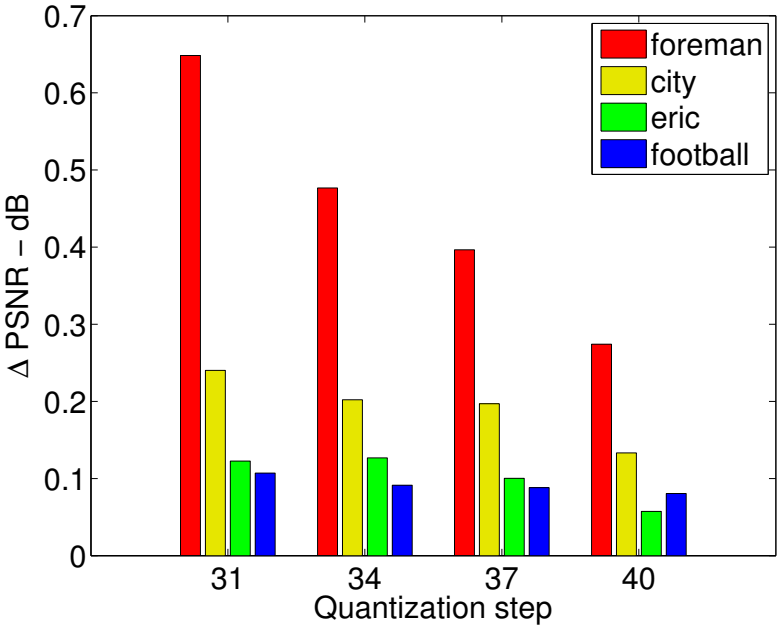


Figure 6.5: SI PSNR differences between *DD* (reference) and *DC* methods.

	QP values			
GOP size	*foreman*	*city*	*eric*	*football*
2	0.65	0.24	0.12	0.11
4	0.46	0.28	0.13	0.11
8	0.39	0.28	0.12	0.12

Table 6.4: SI PSNR improvement of *DC* for different GOP sizes [dB].

2 dB on the single image. The best results have been obtained for the *foreman* sequence, characterized by a complex motion. The gain is still interesting for the sequence *city*, characterized by a more regular motion. Smaller gains are obtained when the movement is more irregular (*football*) and for the sequence *eric*. We observe as in the *CD* tests that the gain is smaller for highly quantized KFs.

A further experiment was conducted in order to assess the efficiency of the *DC* when larger GOP sizes are used. We performed a comparison similar to the one reported in Figure 6.5, the only difference being the distance among the key frames. The results are reported in Table 6.4. It is interesting to observe that the PSNR improvement with respect to the *DD* method is quite consistent even for large GOP sizes.

In the last set of experiments, we used the new SI within the global DVC scheme, and computed the global RD performance for QP=31, 34, 37, 40. This was compared with the RD performance over the test sequences of the reference DISCOVER coder, and the results are again reported using the Bjontegaard metric [Bjontegaard, 2001], for the same four QPs. As shown in Table 6.5, the *DC* method allows some interesting rate reductions (3.5% for foreman and 2.0% in average). The PSNR improvement is smaller than the one we have found on the sole side information. This is reasonable since this time the PSNR is computed on the KFs as well (in order to give a right idea of rate improvement on the whole sequence coding), which are identical for the two schemes.

	foreman	*city*	*eric*	*football*	Average
ΔRate	-3.52%	-1.97%	-1.02%	-1.53%	-2.01%
ΔPSNR	0.18	0.10	0.06	0.08	0.10

Table 6.5: Average RD performances improvement of DC with respect to the reference DD scheme.

6.1.3 Total variation based algorithm

This method was initially developed for inter-camera estimation in stereo vision. To compute the disparity values between two images taken from different viewpoints, the pixels have to undergo a matching procedure, often referred to as the stereo correspondence problem. This process consists in finding for each pixel in one image, its corresponding point in the other image, based on their positions and intensity values. The most critical choice for a stereo matching algorithm is the optimization technique which minimizes a given measure of photometric similarity between pixels.

In the field of dense disparity estimation, global optimization methods have attracted much attention due to their excellent experimental results [Scharstein, Szeliski, 2002]. These methods exploit various constraints on disparity such as smoothness, view consistency etc, while using efficient and powerful optimization algorithms. In this section, we consider a disparity estimation approach based on a set theoretic formulation. The proposed method, described in [Miled *et al.*, 2006] [Miled *et al.*, 2009], is a global stereo method inspired from a work developed for image restoration purposes [Combettes, 2003]. In the adopted set theoretic framework, the main concern is to find solutions that are consistent with all the available information about the problem. Each piece of information, derived from a prior knowledge and consistency with the observed data, is represented by a convex set in the solution space and the intersection of these sets (the feasibility set) constitutes the family of possible solutions. The aim is then to find an acceptable solution minimizing the given objective function. A formulation of this problem in an Hilbert image space $\mathcal{H}$ is therefore:

$$\text{Find } \mathbf{u} \in \mathbf{p} = \bigcap_{i=1}^{m} S_i \text{ such that } J(\mathbf{u}) = \inf J(\mathbf{p}) \ , \tag{6.12}$$

where the objective $J : \mathcal{H} \to (-\infty, +\infty]$ is a convex function and the constraint sets $(S_i)_{1 \leq i \leq m}$ are closed convex sets of $\mathcal{H}$. The constraint sets can generally be modelled as level sets:

$$\forall i \in \{1, \ldots, m\}, \qquad S_i = \{\mathbf{u} \in \mathcal{H} \mid f_i(\mathbf{u}) \leq \delta_i\} \ , \tag{6.13}$$

where, for all $i \in \{1, \ldots, m\}$, $f_i : \mathcal{H} \to \mathbb{R}$ is a continuous convex function and $(\delta_i)_{1 \leq i \leq m}$ are real-valued parameters such that $S = \bigcap_{i=1}^{m} S_i \neq \emptyset$. Many powerful optimization algorithms have been proposed to solve this convex feasibility problem. For the proposed solution, we employ the constrained quadratic minimization method developed in [Combettes, 2003] and particularly well adapted to our needs. However, due to space limitation, we will not describe the algorithm but the reader is referred to [Miled *et al.*, 2006; Combettes, 2003] for more details.

We integrate it in our proposed scheme for both disparity and motion estimation. The next two sections explain how this initial disparity estimation algorithm is adapted to motion or disparity interpolation.

The material in this section was published in:

- W. Miled, T. Maugey, M. Cagnazzo, and B. Pesquet-Popescu, "Image interpolation with dense disparity estimation in multiview distributed video coding," in *Int. Conf. on Distributed Smart Cameras*, Como, Italy, Sep. 2009.
- T. Maugey, W. Miled, and B. Pesquet-Popescu, "Dense disparity estimation in a multi-view distributed video coding system," in *Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, Taipei, Taiwan, Apr. 2009.

6.1.3.1 Monodirectional refinement

6.1.3.1.a Principle

The monodirectional refinement stage aims at improving the forward vectors produced by the monodirectional estimation between the left and right KFs I_a and I_b, using the set theoretic framework described above. For this purpose, we first define the objective function, based on the physical data model. By considering the sum of squared intensity differences (SSD) measure, this objective function can be expressed as follows:

$$\tilde{J}(\mathbf{u}) = \sum_{\mathbf{p}\in\mathcal{D}} [I_a(\mathbf{p}) - I_b(\mathbf{p}+\mathbf{u}(\mathbf{p}))]^2 \tag{6.14}$$

where $\mathcal{D} \subset \mathbb{N}^2$ is the image support. This expression is non-convex with respect to the displacement field $\mathbf{u}$. Thus, in order to avoid a non-convex minimization, we use the initial estimate $\bar{\mathbf{u}}$ produced by the first monodirectional estimation stage (based on a block matching process) and we express the non-linear term $I_b(\mathbf{p}+\mathbf{u}(\mathbf{p})))$ around $\bar{\mathbf{u}}$ using the standard first order approximation:

$$I_b(\mathbf{p}+\mathbf{u}) \simeq I_b(\mathbf{p}+\bar{\mathbf{u}}) + (\mathbf{u}-\bar{\mathbf{u}})\nabla I_b(\mathbf{p}+\bar{\mathbf{u}}) \ , \tag{6.15}$$

where $\nabla I_b(\mathbf{p}+\bar{\mathbf{u}})$ is the gradient of the compensated left frame. Note that for notation concision, we have not made anymore explicit that $\mathbf{u}$ and $\bar{\mathbf{u}}$ are functions of $\mathbf{p}$ in the above expression.

With the approximation of Equation (6.15), the cost function $\tilde{J}$ under the minimization in Equation (6.14) becomes quadratic in $\mathbf{u}$, as follows:

$$J(\mathbf{u}) = \sum_{\mathbf{p}\in\mathcal{D}} [L(\mathbf{p})\ \mathbf{u}(\mathbf{p}) - r(\mathbf{p})]^2 \tag{6.16}$$

where

$$\begin{aligned} L(\mathbf{p}) &= \nabla I_b(\mathbf{p}+\bar{\mathbf{u}}(\mathbf{p})), \\ r(\mathbf{p}) &= I_b(\mathbf{p}) - I_b(\mathbf{p}+\bar{\mathbf{u}}(\mathbf{p})) + \bar{\mathbf{u}}(\mathbf{p})\ L(\mathbf{p}). \end{aligned}$$

Given the objective function to be minimized, we incorporate, in what follows, the constraints modelling prior information on the estimated field as closed convex sets in the form of Equation (6.13). The most common constraint on the field is the knowledge of its range of possible values. Indeed, motion/disparity values often have known minimal and

maximal amplitudes, denoted respectively by $\mathbf{u}^{\min} = (u_x^{\min}, u_y^{\min})$ and $\mathbf{u}^{\max} = (u_x^{\max}, u_y^{\max})$. The associated set is

$$S_1 = \{\mathbf{u} = (u_x, u_y) \in \mathcal{H} \mid u_x^{\min} \leq u_x \leq u_x^{\max} \text{ and } u_y^{\min} \leq u_y \leq u_y^{\max}\} \ . \tag{6.17}$$

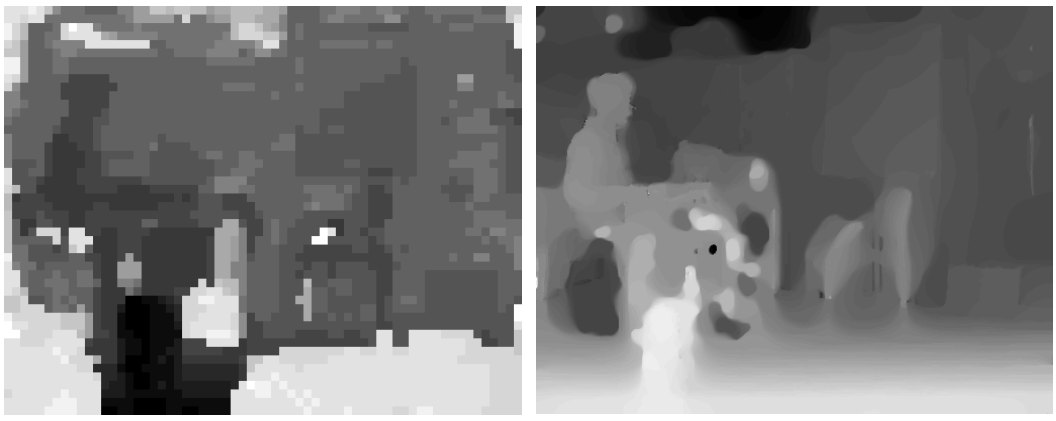
Furthermore, the vector field should be smooth in homogeneous areas while keeping sharp edges. This can be achieved with the help of a suitable regularization constraint. In this work, we make use of the total variation (tv) measure which recently emerged as an effective tool to recover smooth images in various image processing research fields. Practically, $\mathrm{tv}(\mathbf{u})$ represents a measure of the lengths of the level lines in the image [Rudin *et al.*, 1992]. Hence, if $\mathbf{u}$ is known a priori to have a certain level of oscillation so that a bound τ is available on the total variation, controlling $\mathrm{tv}(\mathbf{u})$ restricts the solutions to the convex set

$$S_2 = \{\mathbf{u} \in \mathcal{H} \mid \mathrm{tv}(\mathbf{u}) \leq \tau\} \ . \tag{6.18}$$

It should be noticed that the upper bound τ can be estimated with good accuracy from prior experiments and that the considered minimization method is shown to be robust with respect to the choice of this bound [Miled *et al.*, 2006].

In summary, we formulate the field estimation problem as the minimization of the quadratic objective function (Equation (6.19)) over the feasibility set $S = \cap_{i=1}^{2} S_i$, where the constraint sets $(S_i)_{1 \leq i \leq 2}$ are given by Equations (6.17) and (6.18). The obtained field is then fit into the bidirectional estimation stage to get symmetric predictions from the two KFs.

In practice, the vectors $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ are computed online based on the initial values of the input vector field $\mathbf{u}_{ab}$. The bound τ was set after a set of experiments on several test sequences. The evaluation of the optimal τ value needs to be precise, because too much regularization would prevent taking into account some objects. The value was set to $\tau = 1500$. One can see in Figure 6.6 the effects of the regularization on one example of a disparity field for the rectified video sequence *book arrival*. The refinement algorithm has smoothed the disparity field in the background, and in the objects. However, the contours of the objects keep being sharp.



(a) Initial block-based disparity field, $\mathbf{u}_{ab}$ (b) Refined dense disparity field, $\mathbf{u}_{ab}^*$

Figure 6.6: Visual examples of the difference between block-based and pixel-based horizontal component of the disparity fields for *book arrival* rectified sequence.

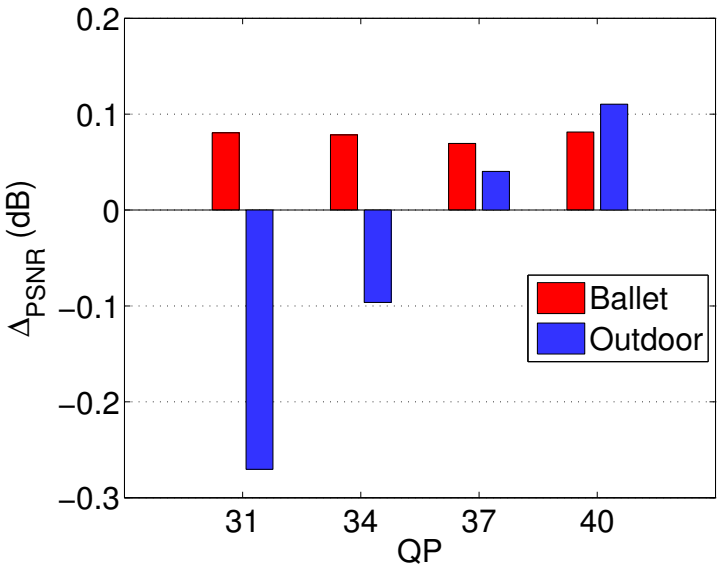


Figure 6.7: SI PSNR differences between DD (reference) and VD methods

6.1.3.1.b First experiments

We evaluate the VD method on the two multiview test sequences *ballet* (non-rectified) and *outdoor* (rectified). For both sequences, the spatial resolution has been halved by two, so that the images have a size of 512×386, and only the first 7 cameras were used. The VD refinement technique has been performed to estimate the SI of the WZFs corresponding to views 2, 4 and 6. For each view, we consider four quantization steps (QP = 31, 34, 37 and 40), in order to compare to the DD algorithm in a relatively wide range of key frame quantization levels.
In Figure 6.7, we plot the average difference between the PSNR of the VD SI and the PSNR of DD SI for these two test sequences. One can see that VD enhances the quality of the SI only for *ballet* and *outdoor* at QP=37 and QP=40. Moreover, this improvement is quite low (less than 0.1 dB). The fall of the PSNR quality for *outdoor* is explained by the fact that the cameras are close, and the initial DD estimation is of a very good quality, and thus hardly improvable (excepting for coarsely quantized key frames, and thus, a lower quality of DD estimation).
However, we can deduce from these first results (further results will be provided in Section 6.1.4) than the VD method does not present for inter-view estimations the same efficiency as Cafforio-Rocca based method for temporal interpolation.

6.1.3.2 Bidirectional refinement

6.1.3.2.a Principle

The bidirectional refinement stage consists in recovering first the forward and backward vectors of the DISCOVER algorithm, denoted respectively by $\mathbf{u}_b$ and $\mathbf{u}_a$, and applying then the iterative optimization algorithm within the set theoretic framework. The cost function to be minimized, in this case, is based on the assumption that the pixel in the image I_b compensated by the forward vector $\mathbf{u}_b^*$ has the same intensity value as the pixel in I_a

compensated by the backward vector $\mathbf{u}_a^*$. It allows to jointly estimate both vectors, as follows:

$$\tilde{J}(\mathbf{u}_a^*, \mathbf{u}_b^*) = \sum_{\mathbf{p}\in\mathcal{D}} [I_a(\mathbf{p} + \mathbf{u}_a^*(\mathbf{p})) - I_b(\mathbf{p} + \mathbf{u}_b^*(\mathbf{p}))]^2. \tag{6.19}$$

This expression is non-convex with respect to the displacement fields $\mathbf{u}_a^*$ and $\mathbf{u}_b^*$. Like in the monodirectional refinement case, it is approximated by first order approximations to get a convex cost function. However, here we expand both I_a and I_b around initial Discover vectors $\mathbf{u}_a$ and $\mathbf{u}_b$, respectively:

$$\begin{aligned} J(\mathbf{u}_a^*, \mathbf{u}_b^*) = \sum_{\mathbf{p}\in\mathcal{D}} & [I_a(\mathbf{p} + \mathbf{u}_a(\mathbf{p})) - I_b(\mathbf{p} + \mathbf{u}_b(\mathbf{p})) \\ & + \nabla I_a(\mathbf{p} + \mathbf{u}_a(\mathbf{p}))(\mathbf{u}_a(\mathbf{p}) - \mathbf{u}_a(\mathbf{p})) \\ & - \nabla I_b(\mathbf{p} + \mathbf{u}_b(\mathbf{p}))(\mathbf{u}_b(\mathbf{p}) - \mathbf{u}_b(\mathbf{p}))]^2 \\ = & \sum_{\mathbf{p}\in\mathcal{D}} [L(\mathbf{p})\mathbf{u}(\mathbf{p}) - r(\mathbf{p})]^2 \ , \end{aligned} \tag{6.20}$$

where we defined

$$\begin{aligned} \mathbf{u} &= (\mathbf{u}_a^*, \mathbf{u}_b^*)^\top \\ L(\mathbf{p}) &= [\nabla I_a(\mathbf{p} + \mathbf{u}_a(\mathbf{p})) - \nabla I_b(\mathbf{p} + \mathbf{u}_b(\mathbf{p}))] \\ r(\mathbf{p}) &= I_a(\mathbf{p} + \mathbf{u}_b(\mathbf{p})) - I_b(\mathbf{p} + \mathbf{u}_a(\mathbf{p})) + L(\mathbf{p})(\mathbf{u}_a, \mathbf{u}_b)^\top. \end{aligned}$$

Once the global convex objective function to be minimized is defined, we add the convex constraints based on the properties of the estimated fields. We retain, as previously, the range values constraint and the edge preserving regularization one. The constraint sets associated with the first *a priori* information are

$$S_1 = \{\mathbf{u} = (u_x, u_y) \in \mathcal{H} \mid u_{ax}^{\min} \leq u_x \leq u_{ax}^{\max} \text{ and } u_{ay}^{\min} \leq u_y \leq u_{ay}^{\max}\} \ , \tag{6.21}$$

$$S_2 = \{\mathbf{u} = (u_x, u_y) \in \mathcal{H} \mid u_{bx}^{\min} \leq u_x \leq u_{bx}^{\max} \text{ and } u_{by}^{\min} \leq u_y \leq u_{by}^{\max}\} \ . \tag{6.22}$$

The regularization constraint, whose effect is to smooth homogeneous regions in the field while preserving edges, introduces a bound on the integral of the norm of the spatial gradient. Thus, imposing an upper bound on the total variation allows to efficiently restrict the solution to the constraint sets:

$$S_3 = \{\mathbf{u} \in \mathcal{H} \mid \mathrm{tv}(\mathbf{u}_a^*) \leq \tau_{\mathbf{u}_a^*}\} \ , \tag{6.23}$$

$$S_4 = \{\mathbf{u} \in \mathcal{H} \mid \mathrm{tv}(\mathbf{u}_b^*) \leq \tau_{\mathbf{u}_b^*}\} \ , \tag{6.24}$$

where $\tau_{\mathbf{u}_a^*}$ and $\tau_{\mathbf{u}_b^*}$ are positive constants that can be estimated from prior experiments and image databases.

The problem of motion/disparity estimation can finally be formulated as jointly finding the forward and backward fields which minimize the energy function in Equation (6.20) subject to the constraints $(S_i)_{1\leq i\leq 4}$. The problem becomes therefore bivariate and to solve it, we have adapted the convex optimization algorithm considered in the monodirectional case, taking into account the dimensionality of the problem.

The parameter τ was experimentally fixed at 1500, which is the same value as in the monodirectional case.

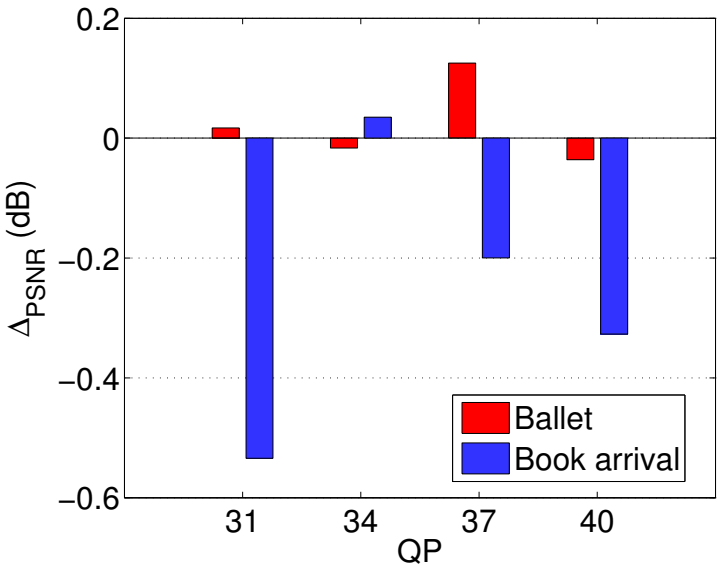


Figure 6.8: SI PSNR differences between DD (reference) and DV methods for inter-view estimations.

6.1.3.2.b First experiments

We evaluate the DV method similarly to the VD one. We consider two multiview sequences (with a resolution halved by two: 512×384) *ballet* and *book arrival* and their first 7 cameras. We calculate the inter-view interpolations with DV and DD methods, for 4 quantization steps for the key frames (QP equal to 31, 34, 37 and 40) and compare their PSNR. Figure 6.8 presents the Δ_{PSNR} results in dB. The efficiency of DV method is obviously disappointing. Indeed, DV does not improve the DD results and even degrades them sensibly for *book arrival* sequence. The total variation based bidirectional refinement seems not to be very efficient for inter-view estimation. In the next section, this method is tested for temporal interpolation.

6.1.4 Experiments

In the previous section, we introduced the proposed refinement methods and tested their integration in the proposed general interpolation scheme for their natural configurations: the Cafforio-Rocca based interpolations were tested for temporal estimation, while the total-variation based methods were applied in inter-view estimations since they are based on Miled's work whose purpose was the disparity estimation. In this section, we propose further experiments where the proposed refinement methods are tested in every configuration (intra and inter-camera) and where they are compared between each other.

With the interpolation scheme proposed in Section 6.1.1 (Figure 6.1), 9 different methods can be considered. The first method is our reference, *i.e.*,Discover. It is referred to as DD. Then, we have seen in the previous sections the one-refinement methods (CD, DC, VD and DV), and we present here more complete tests for them.

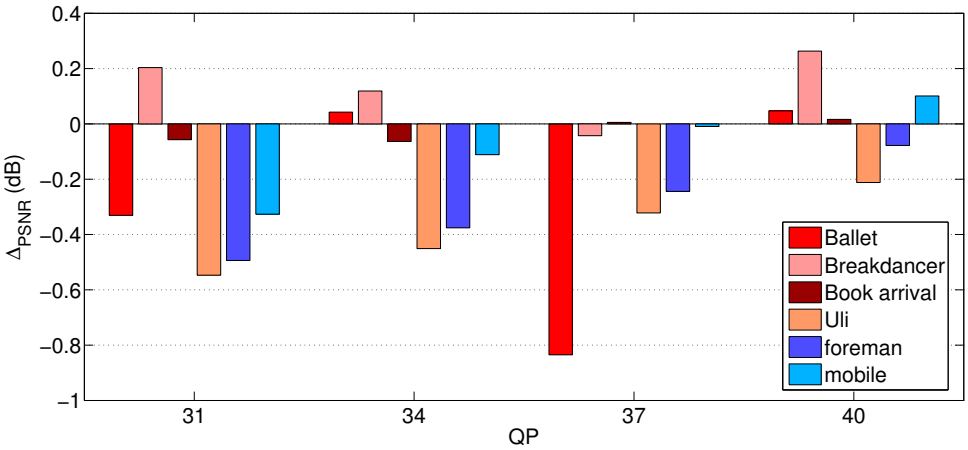


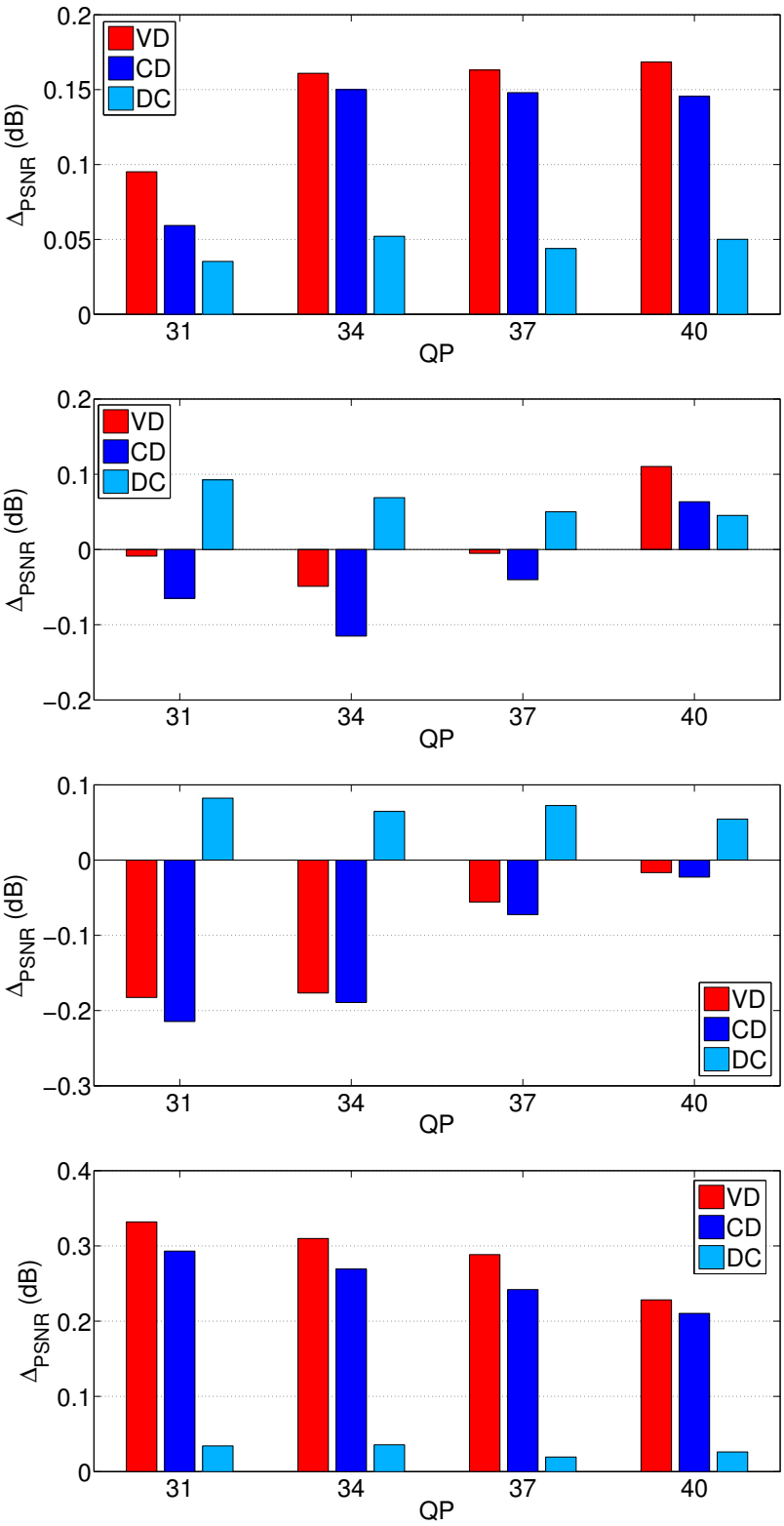
Figure 6.9: SI PSNR differences between DD (reference) and DV methods for temporal interpolations (multiview sequences in red-brown, and monoview ones in blue)

The tests presented in the rest of the section were all obtained under the same experimental conditions. First the reference frames of all the video sequences are intra coded at 4 different QPs: 31, 34, 37 and 40. In the following, when we talk about QP, it corresponds to the quantization of the reference frames[1]. Then, for every QP, we calculate the original method DD in both directions (for multiview sequences only). Then we calculate the interpolation obtained with the CD, DC, VD and DV refinement methods, and compare the PSNR with the DD reference method. Results are shown in figures which plot the Δ_{PSNR} in dB in function of the different QPs.

In our experiments, we observed that the DV method leads to a poorer SI than the Discover interpolation in almost all cases. Results for inter-view estimations have already been given in Figure 6.8. Figure 6.9 shows the performance of the DV temporal interpolation, for several video sequences: multiview sequences in red or brown (*ballet*, *breakdancer*, *book arrival* and *uli*) and monoview sequences in blue (*foreman* and *mobile*). Excepting for *breakdancer* for which DV obtains a quite acceptable improvement for some QPs, the total-variation based monodirectional refinement does not enhance the DD estimations PSNR. That is why, in the following we do not consider this estimation and only compare the three other methods: VD, CD and DC.

In Figures 6.10 and 6.11, we show the Δ_{PSNR} results for 4 multiview sequences of respectively the temporal and the inter-view interpolations. Moreover, we show in Table 6.6 the average Δ_{PSNR} over the frames and over the QPs for several video sequences (monoview and multiview) for the temporal interpolation. One can observe that the proposed methods obtain satisfying performances. Indeed, in temporal direction, the Δ_{PSNR} reaches for example 0.7 dB for *city* sequence, 0.6 dB for *mobile* and 0.3 dB for *outdoor*. However, there is no denying that the presented results are not completely acceptable since

[1]For example, we will say "the method obtains an improvement of ... at a QP of..." instead of "the method obtains an improvement of ... with reference frames quantized at a QP of...".



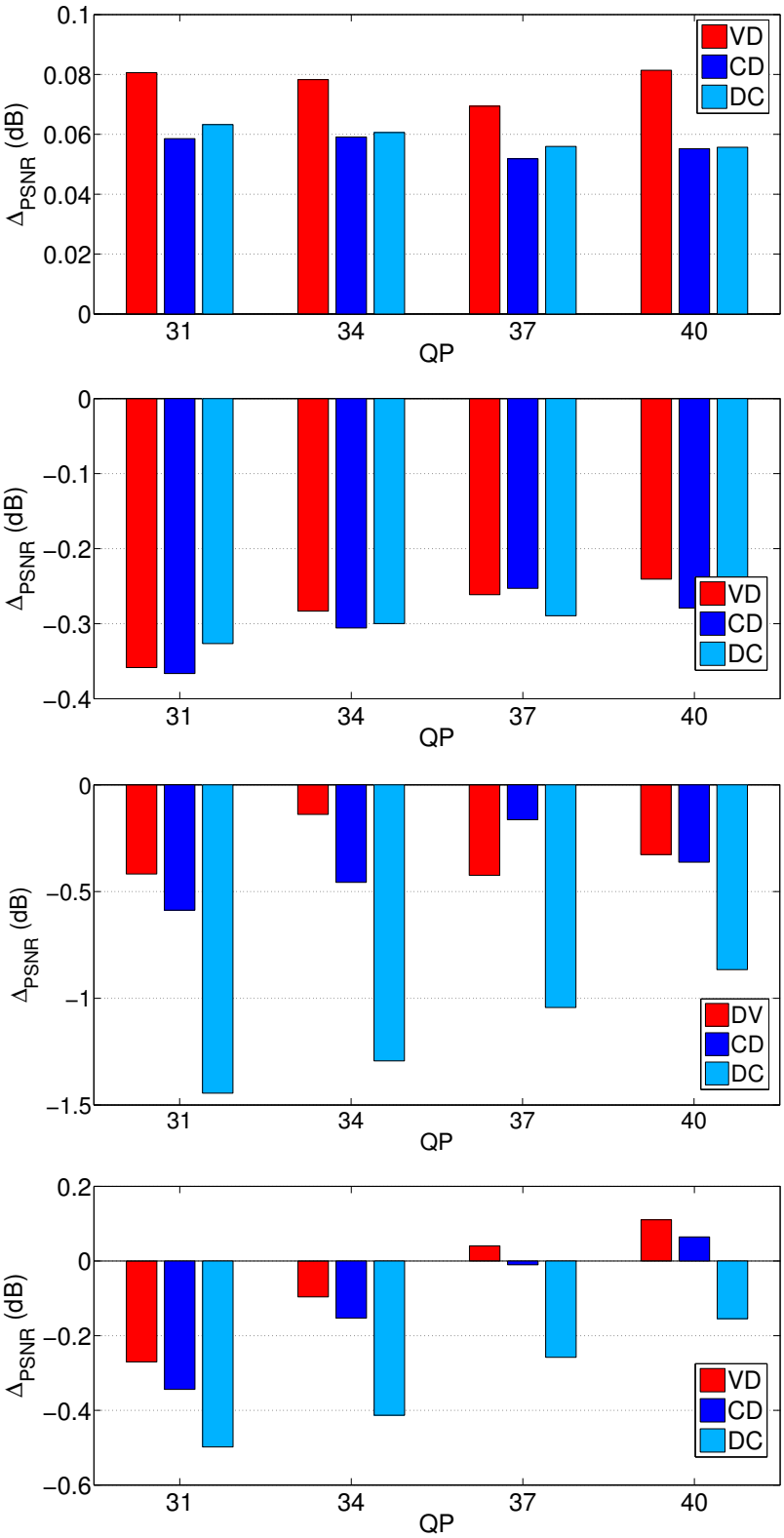
(a) *ballet*

(b) *ballroom*

(c) *book arrival*

(d) *outdoor*

Figure 6.10: Δ_{PSNR} between refinement methods and the reference method DD for temporal interpolation in different multiview sequences.



(a) *ballet*

(b) *ballroom*

(c) *book arrival*

(d) *outdoor*

Figure 6.11: Δ_{PSNR} between refinement methods and the reference method DD for inter-view interpolation in different multiview sequences.

	CD	DC	VD	Mean
akiyo*	0.00	-0.08	0.00	-0.02
city*	0.93	0.17	1.12	0.74
container*	0.17	-0.23	0.17	0.04
eric*	0.18	-0.16	0.24	0.08
football*	-0.27	-0.12	-0.16	-0.18
foreman*	0.20	0.21	0.20	0.21
mother and daughter*	0.01	0.00	0.01	0.01
mobile*	0.84	0.00	1.03	0.62
news*	0.09	0.00	0.09	0.06
tempete*	-0.10	-0.01	-0.08	-0.06
silent*	-0.02	0.02	-0.02	-0.01
waterfall*	0.01	0.01	0.01	0.01
planet* (synthetic sequence)	0.09	0.22	0.14	0.15
book arrival$^+$	-0.12	0.07	-0.11	-0.05
outdoor$^+$	0.25	0.03	0.29	0.19
ballet$^+$	0.13	0.04	0.15	0.11
ballroom$^+$	-0.04	0.06	0.01	0.01
uli$^+$	-0.00	0.03	0.02	0.02
Mean	**0.13**	**0.02**	**0.17**	**0.11**

Table 6.6: Average SI Δ PSNR in temporal direction for several test sequences. *: monoview sequences (352×288, 30 fps), $^+$: multivew sequences (512×384, 30 fps)

they are limited sometimes (ex: *waterfall*, *ballet* in the view direction) and even negative in some cases (ex: *football*, *ballroom* in view direction). Nevertheless, the Δ_{PSNR} drawn in Table 6.6 present promising aspects. Indeed, the average improvement is positive and around 0.13 for CD, and 0.17 for VD for temporal interpolation. One can observe that the refinement methods are more competitive for temporal estimations. In this configuration it is interesting to observe that the DC method leads to very limited gain, almost every time lower than 0.1 dB. On the other hand, monodirectional refinements VD and CD sometimes sensibly improve the temporal DD estimations (*ballet* and *outdoor*) but sometimes degrade it (*ballroom* and *book arrival*).

Moreover, we notice that the monodirectional refinements methods are more efficient than the bidirectional ones. It can be explain by the fact that the monodirectional refinement is followed by several steps which have a better behaviour if their initialization is more precise and reliable, as it is the case with CD and VD methods. This is the reason why we have not investigated the double-refinement methods (CC, CV, VC, VV). Indeed, we have reached the best improvements for the monodirectional refinements, but almost 0 dB in average for bidirectionnal one, therefore it appeared hopeless to perform both refinements.

However, though the monodirectional refinement methods seem to build side information of better quality, they sometimes sensibly degrade the DD interpolation. This could be explained by the fact that these methods strongly depend on their parameter optimization. Indeed, they have been optimized for some videos, as it was explained in the

previous sections, and these parameters were kept for the other sequences of the database. The parameters are thus not optimal anymore, and this explain why the methods are less efficient. Such a parameter dependency would be a main drawback of our method, unless further works would lead to an online optimal parameter estimation.

6.2 Proposed fusion methods

The material in this section was published in:

- T. Maugey, W. Miled, M. Cagnazzo, and B. Pesquet-Popescu, "Fusion schemes for multiview distributed video coding," in *Proc. Eur. Sig. and Image Proc. Conference*, Glasgow, Scotland, Aug. 2009.

6.2.1 Recall of the context

Another step in the side information construction is the merging of several estimations in the multiview setting. This fusion is mainly performed at the pixel level in the literature. In this section we propose some other dense fusions methods. We adopt the same notations as those which were introduced in Part II, Section 4.2.2. They are recalled in Figure 6.12. For the estimation of a WZ frame W, four images are available, which are used to generate four motion/disparity compensated frames.

6.2.2 Proposed techniques

The fusion solutions presented in the side information generation state-of-the-art chapter (Section 4.2.2) section achieve good performance in some cases. For example, the PD (Pixel difference) fusion is quite efficient when the temporal motion activity is low. On the contrary, non-fusion estimation qualities strongly depend on the sequence. In this section, we propose three new methods aiming at more robustness. The first two use the residual (*i.e.* the difference between the two compensated reference frames), like the MCD fusion does. The residual is commonly used to approximate the estimation error in DVC, for example for the distribution model analysis at the turbo decoder.

The **motion and disparity compensated difference binary fusion** (MDCDBin) compares the temporal and inter-view residuals, and uses for the estimation the one having the smallest one at each position. As for the existing solutions, the decision is binary. The temporal and inter-view residuals are respectively defined as $E_T(\mathbf{p}) = |\tilde{I}_{n,t^-}(\mathbf{p}) - \tilde{I}_{n,t^+}(\mathbf{p})|$ and $E_N(\mathbf{p}) = |\tilde{I}_{n^-,t}(\mathbf{p}) - \tilde{I}_{n^+,t}(\mathbf{p})|$. Therefore, the prediction by MDCDBin is defined as:

$$\tilde{I}(\mathbf{p}) = \begin{cases} \tilde{I}_N(\mathbf{p}), & \text{if } E_N(\mathbf{p}) < E_T(\mathbf{p}) \\ \tilde{I}_T(\mathbf{p}), & \text{otherwise.} \end{cases}$$

This criterion is improved in the case of **motion and disparity compensated difference linear fusion** (MDCDLin), where the residuals E_T and E_N are no longer used to take a binary decision, but rather to compute a linear combination of inter-view and temporal estimations. The prediction by MDCDLin is then:

$$\tilde{I}(\mathbf{p}) = \frac{E_T(\mathbf{p})}{E_T(\mathbf{p}) + E_N(\mathbf{p})}\tilde{I}_N(\mathbf{p}) + \frac{E_N(\mathbf{p})}{E_T(\mathbf{p}) + E_N(\mathbf{p})}\tilde{I}_T(\mathbf{p})$$

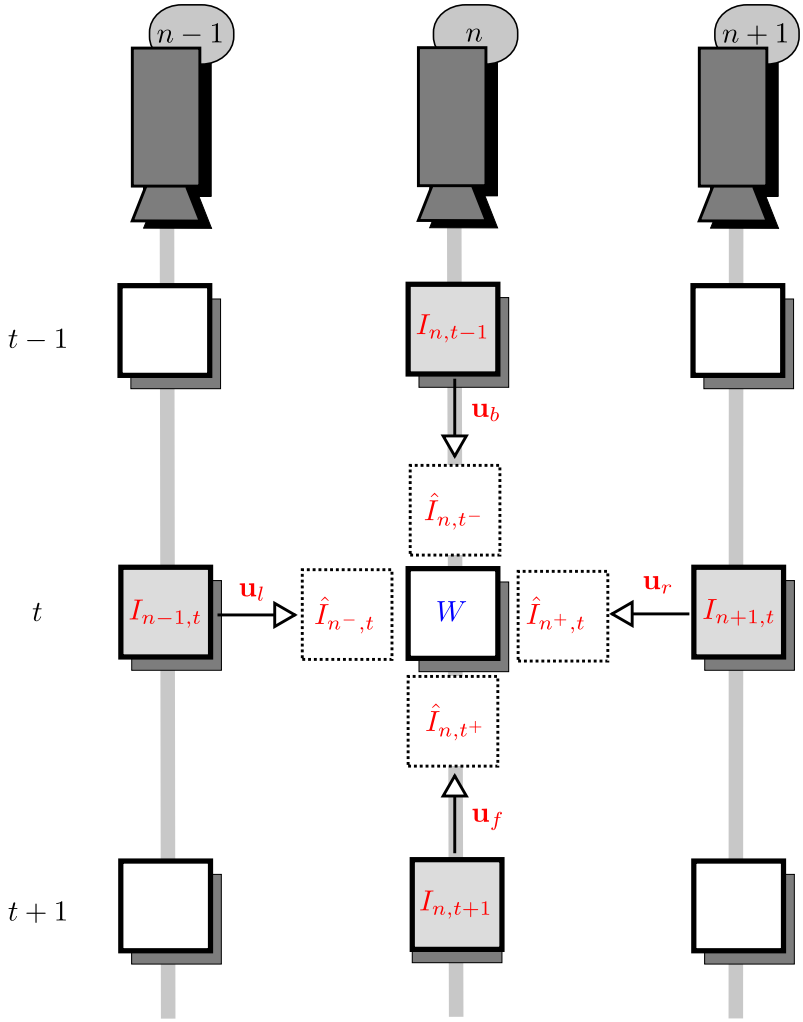


Figure 6.12: Fusion problem: I_x are the available KFs and $\hat{I}_x$ their motion compensated version, estimating the WZ frame W. $\mathbf{u}_x$ are the vector fields.

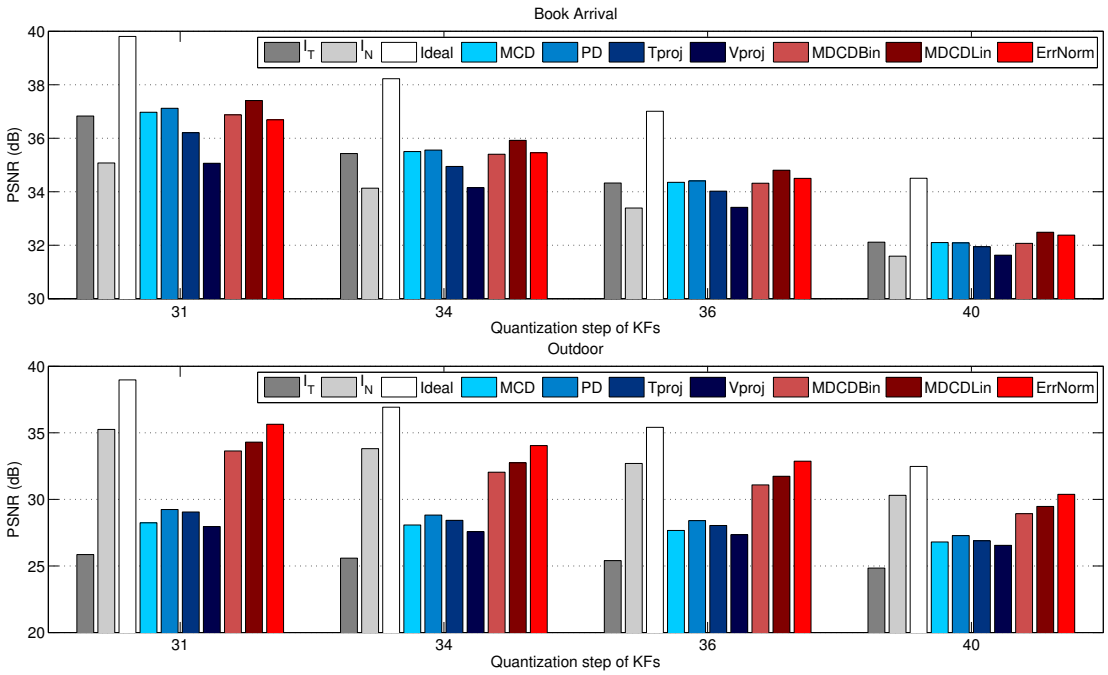


Figure 6.13: SI quality for different fusion methods, at different KF quantization levels, and for two test sequences *book arrival* and *outdoor*.

Finally, in the case of **Estimation-error and vector-norm based linear fusion** (ErrNorm), we build on the consideration that often the larger are the motion vectors, the less reliable is the estimation. Therefore, we use the motion vector norms as weights in computing a linear combination between $\tilde{I}_T$ and $\tilde{I}_N$. The resulting image is then averaged with the one produced by MDCDLin to obtain the new estimation. More precisely, in the ErrNorm case we have the following equations:

$$\tilde{I}(\mathbf{p}) = \frac{\tilde{I}_{\text{err}}(\mathbf{p}) + \tilde{I}_{\text{norm}}(\mathbf{p})}{2} \qquad \text{where}$$

$$\tilde{I}_{\text{norm}}(\mathbf{p}) = \frac{(\|\mathbf{v}_b\| + \|\mathbf{v}_f\|)\tilde{I}_N(\mathbf{p}) + (\|\mathbf{v}_l\| + \|\mathbf{v}_r\|)\tilde{I}_T(\mathbf{p})}{\|\mathbf{v}_b\| + \|\mathbf{v}_f\| + \|\mathbf{v}_l\| + \|\mathbf{v}_r\|}$$

$$\text{and} \quad \tilde{I}_{\text{err}}(\mathbf{p}) = \frac{E_T(\mathbf{p})\tilde{I}_N(\mathbf{p})}{E_T(\mathbf{p}) + E_N(\mathbf{p})} + \frac{E_N(\mathbf{p})\tilde{I}_T(\mathbf{p})}{E_T(\mathbf{p}) + E_N(\mathbf{p})}$$

6.2.3 Experimental results

We compared the state-of-the-art fusion techniques presented in Section 4.2.2 with the proposed ones, by running them on two multiview test sequences, *book arrival* and *outdoor*, from [Ingo Feldmann *et al.*, 2008]. For both sequences, the spatial resolution was halved from 1024×772 to 512×386, and only the first 8 cameras were used. We performed the dense WZ frame estimation algorithm in order to produce the vector fields for both temporal and inter-view interpolations. We considered lossy coded KFs and four quantization steps (QP= 31, 34, 36 and 40), in order to observe the behavior of fusion methods in a relatively wide range of bit-rates.

The performance of all the methods are shown in Figure 6.13, where we give the

QP	31	34	36	40
PD	-6.0131	-4.9926	-4.2939	-3.0226
MCDLin	-0.9516	-1.0624	-0.9639	-0.8322
ErrNorm	0.3893	0.2253	0.1658	0.0740

Table 6.7: Δ_{PSNR} between different fusion method and the best non-fusion estimation (inter-view estimation in this case) for *outdoor* sequence.

QP	31	34	36	40
PD	0.2901	0.1293	0.0807	-0.0244
MCDLin	0.5777	0.4926	0.4799	0.3709
ErrNorm	-0.1393	0.0271	0.1761	0.2636

Table 6.8: Δ_{PSNR} between different fusion method and the best non-fusion estimation (temporal estimation in this case) for *book arrival* sequence.

PSNR of the SI with respect to the original WZF. Gray bars correspond to simple cases, where only temporal or inter-view estimation are considered, the white bar corresponds to the ideal (i.e. oracle-driven) fusion, the blue bars are the state-of-the-art methods explained in Section 4.2.2, and the red ones are the proposed techniques. We notice that for the *book arrival* test sequence, the temporal estimation is slightly better than the inter-view one, while the opposite is true for the second sequence, *outdoor*. In both cases, the comparison between the ideal fusion (which can be seen as an upper bound for fusion method performances) and no-fusion cases, shows that fusion can sensibly improve the WZF estimation.

However state-of-the-art methods look like not being able to adequately take advantage from the fusion: while for the *book arrival* sequence, MCD and PD fusions obtain good performances, much better than the non-fusion predictions $\tilde{I}_T$ and $\tilde{I}_N$, this is no longer the case for the second sequence, where state-of-the-art methods perform worse than simple inter-view estimation. We conclude that these methods are not robust enough when there is a sensible gap of quality between the temporal and inter-view estimations.

Different observations can be made for the proposed methods (red bars in Figure 6.13). The first remark is that MDCDLin outperforms MDCDBin, showing that a linear based fusion is more efficient than a binary decision based method. Moreover, for the *book arrival* sequence, the MDCDBin method reaches better performances than the existing solutions. For the *outdoor* sequence, where the other solutions obtain a lower SI quality, the proposed methods achieve good results and ErrNorm fusion sensibly improves the $\tilde{I}_N$ prediction. Finally, for ease of comparison, some of the results in Figure 6.13 are reported in Tables 6.7 and 6.8, in terms of the difference between the best non-fusion estimation for each sequence and three fusion methods, PD (the best existing method), MDCDLin and ErrNorm (the best proposed methods).

In Figure 6.14 we present the rate-distortion performance obtained when using PD, MDCDLin and ErrNorm within a complete DVC multiview coder (inspired by DISCOVER [Areia *et al.*, 2007]). The results confirm that the proposed methods (red curves) outperform existing ones (blue curves). In order to facilitate the comparison, the average

	Δ Rate (%)	Δ PSNR (dB)
PD	21.96	-0.84
MCDLin	2.24	-0.13
ErrNorm	-3.64	0.22

Table 6.9: Rate-distortion performance comparison between the different fusion methods and the inter-view non-fusion estimation for *outdoor* sequence, obtained with the Bjontegaard metric [Bjontegaard, 2001].

	Δ Rate (%)	Δ PSNR (dB)
PD	-2.78	0.19
MCDLin	-6.07	0.37
ErrNorm	-3.13	0.20

Table 6.10: Rate-distortion performance comparison between the different fusion methods and the temporal non-fusion estimation for *book arrival* sequence, obtained with the Bjontegaard metric [Bjontegaard, 2001].

performances computed with the Bjontegaard metric [Bjontegaard, 2001] are shown in Tables 6.9 and 6.10. We note that ErrNorm is consistently better than the non-fusion techniques (obtaining a rate reduction up to 3.83%), while MDCDLin is always better than PD, which in turn, is much worse than the non-fusion method for the *outdoor* sequence.

6.3 Conclusion

In this chapter we have investigated the interest of adopting a pixel approach for the side information generation. Based on several experiments, we have highlighted the potential of dense estimation and fusion. Whereas the proposed interpolation methods are not yet optimized since they do not lead to a systematic improvement, they already show promising results, a mean improvement of 0.11 dB over 19 test sequences. On the other hand, the proposed fusion techniques seem to be more stable than the existing methods.

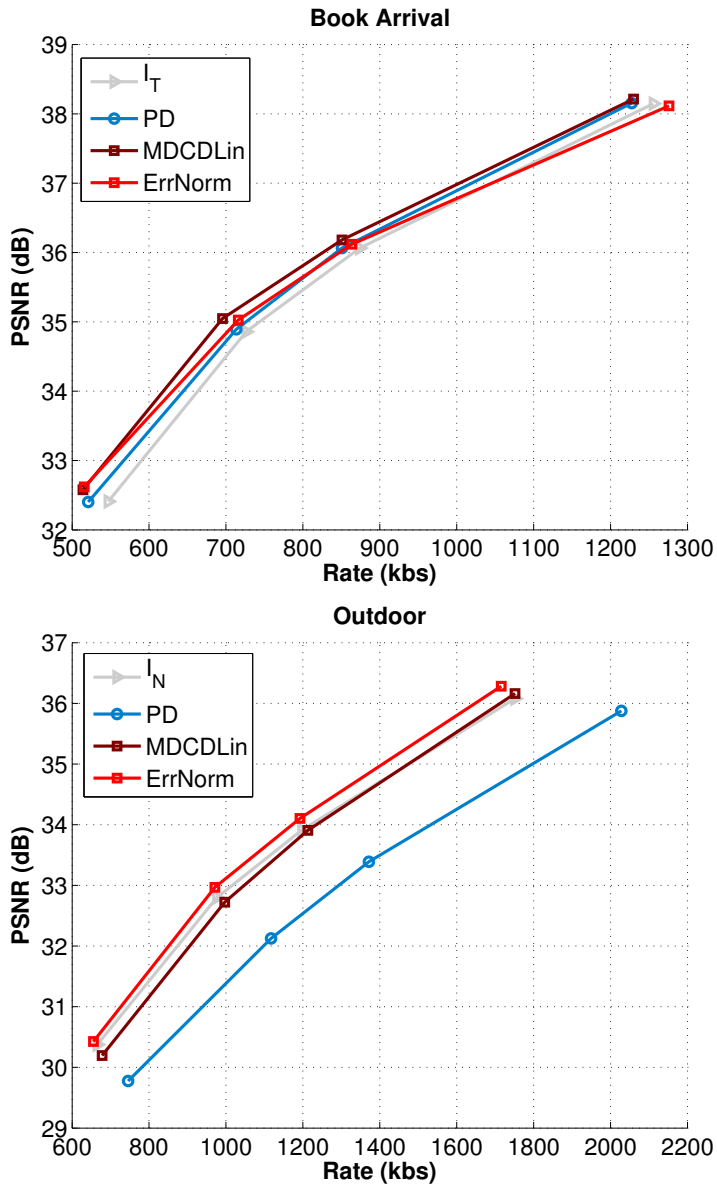


Figure 6.14: Rate-distortion performances for three fusions methods and the best non-fusion estimation, for *outdoor* and *book arrival* test sequences.

Chapter 7

Hash-based side information generation

In some situations (occlusion, rapid motion, etc.), SI generation is limited since the information to be estimated is hardly predictable (limited displacement model, lack of information in the reference frames, etc.). Distributed video coding schemes have to modify their approach for enhancing the WZ estimation at the decoder. We have seen in Section 4.3.3 that some schemes adopt a hash-based approach, in which the encoder sends to the decoder well chosen WZ information (intra coded) in order to facilitate the side information generation and then to enhance the efficiency of channel decoding.
In this chapter, we present a novel hash based scheme mainly inspired by Yaccoub's work [Yaacoub et al.*, 2009a; Yaacoub* et al.*, 2009b; Yaacoub* et al.*, 2009c]. We recall here that Yaacoub et al. have investigated how to enhance the side information quality in monoview DVC by performing a genetic algorithm (GA) based fusion, but without studying precisely the selection and encoding of hash information. We propose here to extend their work by constructing a complete hash-based scheme with an original hash selection and compression. Moreover the proposed scheme is tested in monoview and multiview conditions. First in Section 7.1, we introduce the general structure of the proposed scheme. Then in Section 7.2, we make a zoom on some specific steps of the proposed algorithm where the configuration (monoview/multiview) impact on the developed techniques. Finally, in Section 7.3, we present the experimental results of the proposed hash-based scheme.*

Contents

7.1 Proposed algorithm 172
7.1.1 General structure 172
7.1.2 Hash information generation 173
7.1.3 Genetic algorithm 175
7.2 Zoom on the three setting-dependent steps 175
7.2.1 Initial side information generation 175
7.2.2 Side information block distortion estimation 177
7.2.3 Candidates of the Genetic Algorithm 177
7.3 Experimental results 179
7.3.1 First results 179
7.3.2 Rate-distortion results 180
7.4 Conclusion 180

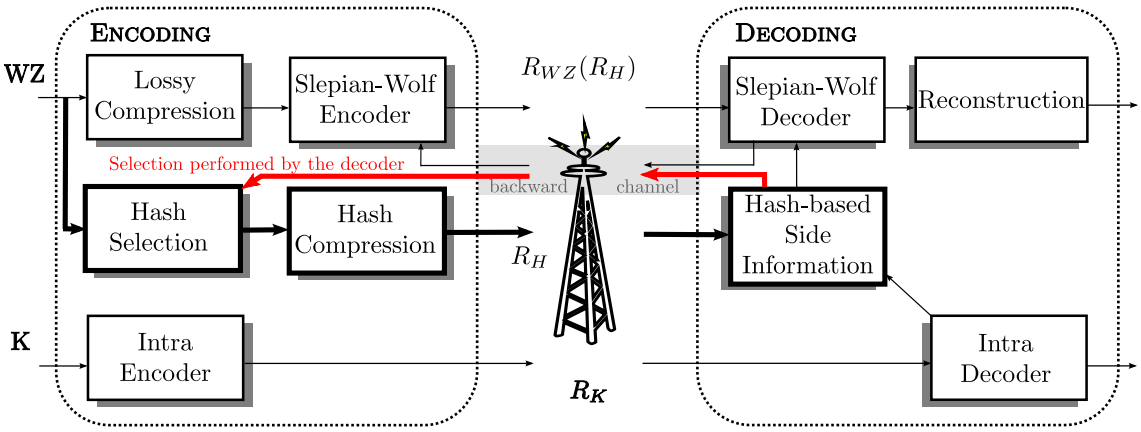


Figure 7.1: General structure of the hash-based DVC scheme. In red, the specificity of our proposed solution, in which the hash-based selection is performed at the decoder.

7.1 Proposed algorithm

The algorithm presented here proposes to improve the side information quality using some hash information sent by the encoder to perform a fusion based on a genetic algorithm. The general structure of a hash-based scheme is summarized in Figure 7.1. As we have seen in Section 4.3, hash-based schemes have to deal with three main issues.

Firstly, the hash information has to be cleverly selected. More precisely, the encoder needs to guess exactly where the decoder would fail in the WZ estimation. This step is fundamental, since the hash information is very expensive in terms of rate cost. State-of-the-art approaches perform this selection at the encoder, by coarsely estimating the side information (average between the reference frames) and thresholding the difference with the true original error. In our approach, we have chosen to perform this selection at the decoder (red arrow in Figure 7.1). In spite of the fact that the original frame is not available anymore at the decoder, the hash selection module has access to the exact WZ estimation knowledge. We believe that the knowledge of the side information at the decoder is more interesting and useful than the knowledge of the original frame at the encoder with a poor available estimation of the SI. Secondly, the hash has to be compressed and transmitted to the decoder. Thirdly, hash information is used at the decoder to generate a finer SI. The proposed approach is based on a fusion of several estimations, contrary to state-of-the-art methods which only perform a hash motion interpolation, as it was explained in Section 4.3.

The general structure of our proposed hash selection and hash-based side information generation algorithm is presented in Section 7.1.1, and then a zoom on the hash information coding and the genetic algorithm are proposed respectively in Section 7.1.2 and Section 7.1.3.

7.1.1 General structure

The general structure of the proposed system at the decoder is presented in Figure 7.2. The method consists in firstly generating a classical side information and secondly, for each badly-estimated block, requesting some hash information from the encoder in the meantime

as the parity bits for turbodecoding so that a hash-based side information estimation can be performed at the decoder side. Therefore, unlike the previous works [Ascenso, Pereira, 2007] and [Aaron *et al.*, 2004a], the intraframe encoding paradigm is preserved here, since the decision on the need of sending hash information is performed at the receiver, instead of thresholding the difference between the two reference (key) frames. The different steps of our hash-based side information generation algorithm (at the decoder side) are as follows:

1. **SI construction -** The encoder generates a side information using the available neighboring reference frames. The adopted technique depends on the configuration (monoview or multiview). The obtained SI is divided into several 4×4 blocks, referred to as b_k^{SI}, and each block is processed independently by the subsequent operations of the algorithm.

2. **SI quality -** For each block, the distortion of the side information is estimated at the decoder side. Similarly to "SI construction" step, the adopted technique depends on the number of available reference frames. The distortion is denoted by D_k for the block b_k^{SI}.

3. **Thresholding -** The D_k value is thresholded by a T value which is calculated depending on the percentage of hash blocks sent to the decoder. If the distortion is lower than T, the side information is considered good enough, such that it can be directly turbo-decoded. Otherwise, b_k^{SI} is assumed to be a bad estimation of the original WZ frame, and therefore the *Hash SI construction* is performed.

4. **Hash SI construction -** The side information is re-estimated thanks to some hash information transmitted at a rate of r_k^H. First, several estimations are generated depending on the scenario (monoview/multiview). Then, the GA algorithm is performed in order to build the fusion of these candidates. The computed hash-based side information b_k^{HSI} depends on the rate r_k^H.

5. **Block assembling -** It consists in constructing the entire side information by assembling the blocks estimated with (b_k^{HSI}) or without (b_k^{SI}) the hash information. The final side information (FSI) is then turbo-decoded.

7.1.2 Hash information generation

The block size is fixed to 4×4, the same as the DCT block size of the Wyner Ziv frame coding. In the following, we describe how each hash block (a vector of 16 coefficients, one per band) is encoded. Based on the fact that some information regarding the WZ frame (as the dynamic range of the bands) is available at the decoder, we decide to perform uniform quantization of the hash information, similar to the quantization performed for WZFs encoding (with a dead zone for the AC coefficients). The number of quantization levels is specified by a quantization matrix, showing the number of levels per band for eight rate-distortion points (from low to high bit rate). The matrix [Brites *et al.*, 2006b] is recalled in Table 7.1.

After the quantization process, the hash is converted into bitplanes and transmitted to the decoder. The corresponding rate is given by the sum of the logarithms of the non-zero bands levels at a chosen line of the quantization matrix.

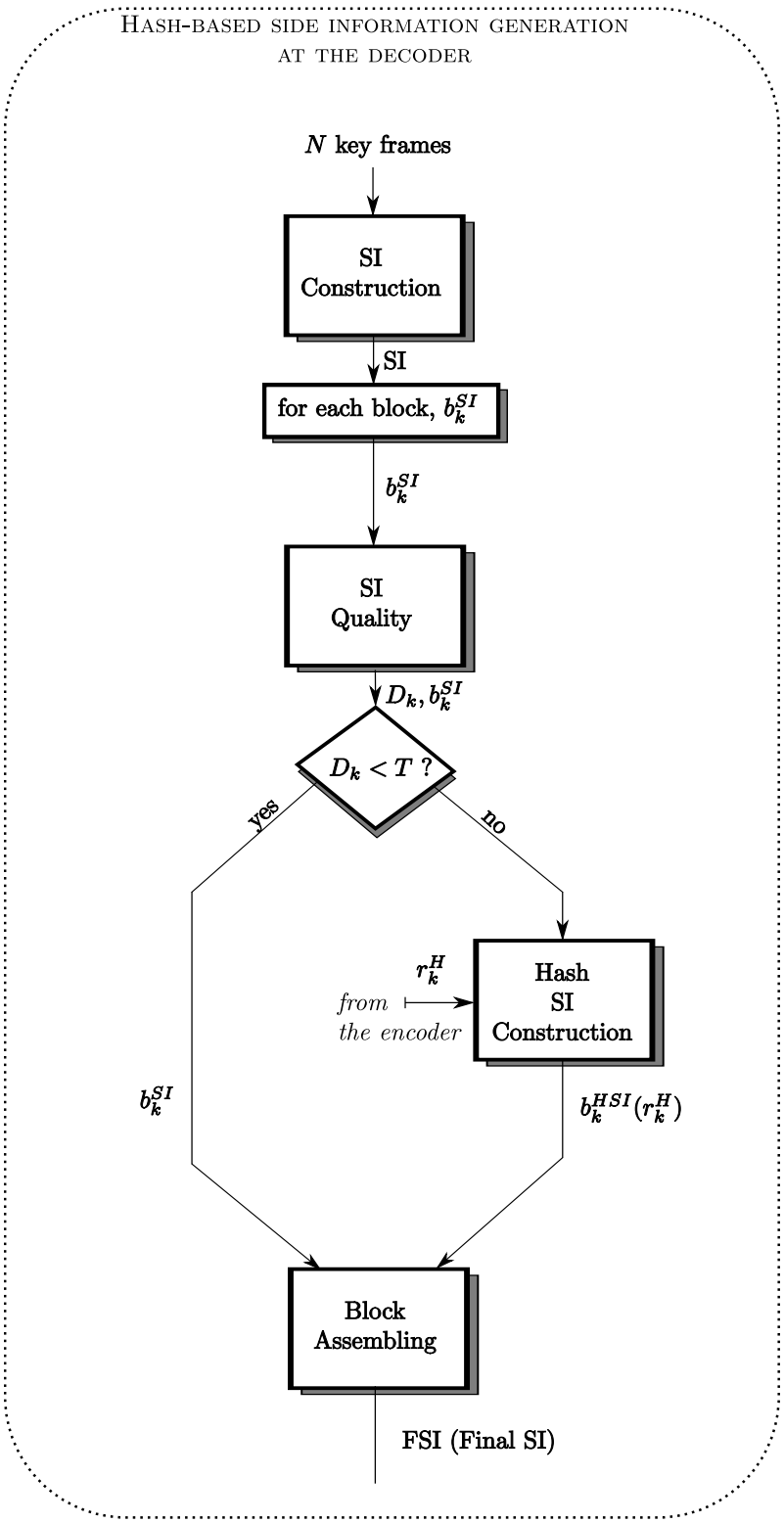


Figure 7.2: General structure of the hash-based side information generation algorithm performed at the decoder side.

Table 7.1: WZ and hash quantization matrix

band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
QI 1	16	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
QI 2	32	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
QI 3	32	8	8	4	4	4	0	0	0	0	0	0	0	0	0	0
QI 4	32	16	16	8	8	8	4	4	4	4	0	0	0	0	0	0
QI 5	32	16	16	8	8	8	4	4	4	4	4	4	4	0	0	0
QI 6	64	16	16	8	8	8	8	8	8	8	4	4	4	4	4	0
QI 7	64	32	32	16	16	16	8	8	8	8	4	4	4	4	4	0
QI 8	128	64	64	32	32	32	16	16	16	16	8	8	8	4	4	0

7.1.3 Genetic algorithm

As explained before, for some part of the SI, the decoder uses a GA algorithm for an hash-based raffinement of the WE frame estimation process. A flowchart diagram of this GA is shown in Figure 7.3. The GA operates at the block level. Initially, for a given block in the WZ frame, each of the co-located blocks in the available SI candidate frames represents a possible solution. A candidate solution is referred to as a *chromosome*, which consists of a sequence of pixels (*genes*) arranged in a matrix to form a block. A population is a set of chromosomes in the solution space. The similarity between a given chromosome and the corresponding block in the WZ frame represents its *fitness* score, which is evaluated as the inverse of the mean-square-error between the received hash word and a local hash word extracted from the candidate block.
An initial population is first generated by duplicating each candidate block a number of times proportional to its fitness, until the desired population size S_p is reached. The chromosomes are then randomly shuffled and arranged into pairs. Each pair (parent chromosomes) undergoes a vertical crossover followed by an horizontal crossover to yield a couple of child chromosomes (called *offsprings*). Each of the crossover operations occurs with a probability P_c. In order to extend the solution space and reduce the possibility of falling into local optima, a mutation is performed on offsprings by randomly selecting a gene and inverting one of its bits. Mutation usually has a very low probability of occurrence P_m [Chang *et al.*, 2001]. The fitness of the resulting chromosomes is then evaluated and a number $S_f \leq S_p$ of the most fit chromosomes is selected, while the others are deleted to make room for new ones. The surviving chromosomes are then duplicated a number of times proportional to their fitness and the whole procedure is repeated until the maximum number I_{max} of iterations is reached. Finally, the fittest chromosome is chosen as the best candidate to be used as side information for decoding the colocated block in the WZ frame.

7.2 Zoom on the three setting-dependent steps

Some of the steps presented in the algorithm above change whether they are involved in a monoview or a multiview scheme. More precisely, as long as the block needs the reference frames around, the adopted method would be different wether they are involved in a monoview or multiview configuration.

7.2.1 Initial side information generation

The proposed hash SI algorithm is based on a first SI estimation. This WZ estimation is based on the available reference frames. The adopted method depends on the number of reference frames used for the SI generation process.

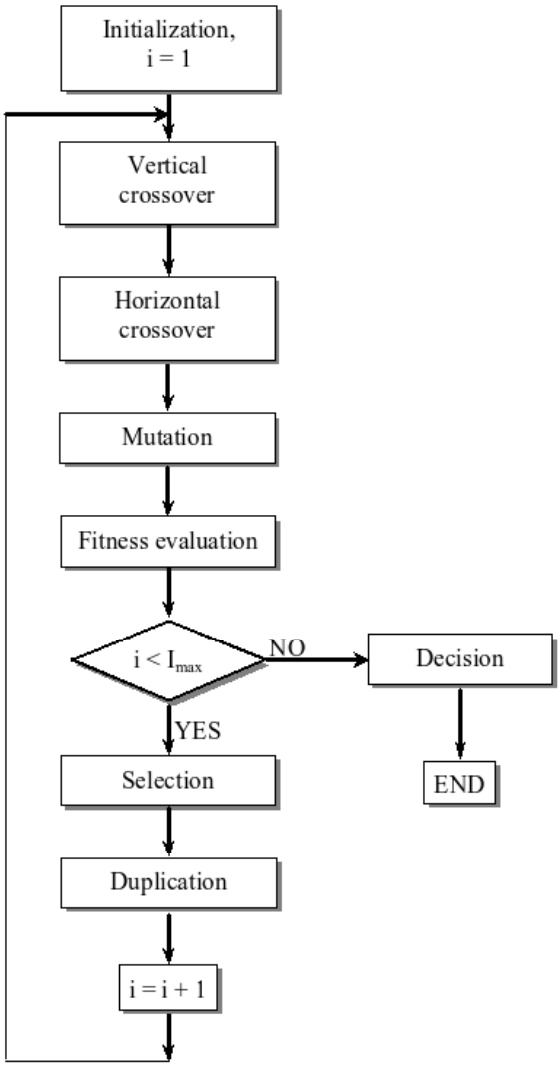


Figure 7.3: Flowchart diagram of the genetic algorithm.

Monoview: for a one-view setting, we generate the initial side information with an interpolation algorithm. More precisely we use the efficient DISCOVER interpolation technique. The reader can refer to Section 4.1.1 for more details.

Multiview: in a multi-camera configuration, more than two frames are available. More precisely, this number depends on the adopted scheme (or frame type repartition in the time-view space). The hash-based scheme is assumed to be integrated in a scheme named symmetric $\frac{1}{2}$ (see Section 3.1), where the type of frames is distributed as a chessboard in the time-view space. Therefore, for the estimation of one WZ frame, four reference frames are available. Two of them belong to the same camera as the WZ frame and are used to generate a temporal interpolation (DISCOVER). The two others belong to the neighboring cameras at the same instant as the estimated WZ frame. They are used to generate an inter-view interpolation (DISCOVER). The two interpolations are then merged using the proposed ErrNorm fusion method (see Section 6.2).

7.2.2 Side information block distortion estimation

For each block of the generated side information estimation, the decoder needs to estimate the distortion without using the original frame. We propose here two approaches (for monoview and multiview settings) which are based on the reference frames, and the previously estimated motion/disparity vector fields.

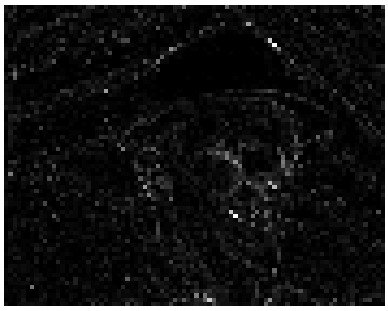
Monoview: The technique used for this SI distortion estimation is the mean square of the difference between the two motion-compensated reference frames, a technique usually adopted for estimating the distortion while performing estimation fusion (see Section 4.2). This approach works under the hypothesis that in the regions where the two motion compensated frames differ, the SI would be badly estimated, and on the contrary, the fact the two motion compensated reference frames are similar would signify that the SI estimation is reliable. A visual result is shown in Figures 7.4 (a) and (b). One can see that the transmitted hash blocks actually correspond to the regions where the side information has important errors.

Multiview: The multiview approach is quite similar. Indeed, the decoder firstly performs the difference between the motion/compensated reference frames (of the same camera), and secondly the difference between the disparity compensated reference frames (of the neighboring cameras). Then these two errors are combined using the coefficients of the linear interpolation fusion ErrNorm (see Section 6.2). As for the monoview setting, we compare in Figures 7.4 (a) and (b) the true error of the WZ estimation and the selected hash blocks. One can see that the blocks are transmitted for regions where high errors occur.

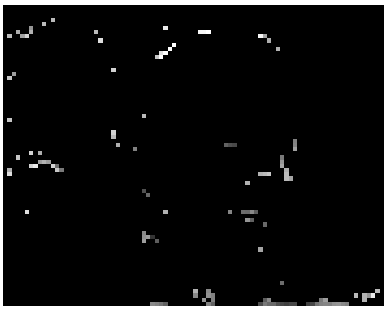
7.2.3 Candidates of the Genetic Algorithm

The genetic algorithm aims at merging a certain number of candidates. These candidates are obtained by using different estimation methods (mainly interpolation).

Monoview: we propose to use the same set of candidates as in the original genetic algorithm based fusion proposed by Yaacoub et al. [Yaacoub *et al.*, 2009a]: an average between the two reference frames, a simple Motion-Compensated Interpolation (MCI) [Aaron



(a) SI error (*foreman*)



(b) hash blocks sent (*foreman*)

(c) SI error (*outdoor*)

(d) hash blocks sent (*outdoor*)

Figure 7.4: Comparison between the true error and the selected hash blocks for *foreman* and *outdoor* sequences.

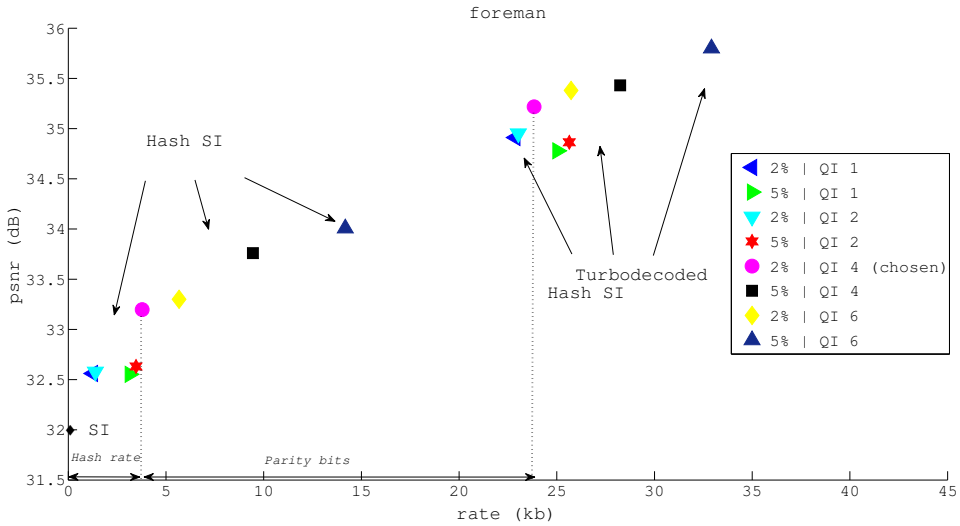


Figure 7.5: Tests for several parameter settings (*foreman*, 352 × 288): percentage of hash information sent from 2% to 5%, and QI from 1 to 6. The best configuration is (2% | QI 4).

et al., 2002] and the Hash-based MCI (HMCI) [Aaron *et al.*, 2004a]. Moreover we propose to add the DISCOVER interpolation.

Multiview: for multi-camera configuration, we propose to use all of the dense interpolation methods proposed in the previous chapter (CD, DC and VD). Each of these three techniques generates one interpolation in the temporal direction, and one inter-view estimation (*i.e.*,6 candidates). Moreover, each couple of temporal/inter-view estimations is merged in order to generate three other candidates. The adopted fusion is again the ErrNorm, because it is competitive and because it performs linear combination between the pixels, and then creates real new candidates (compared to binary fusions which would have given only a duplicate of existing candidates).

7.3 Experimental results

The results presented here have been obtained with three CIF (352 × 288) test sequences (*foreman*, *mobile* and *football*) for a GOP size of 2 (for monoview setting) and another test sequence for multiview configuration: rectified *outdoor* (512 × 384).
For the GA parameters, the following set was determined experimentally after intensive simulations: $S_p = 60$, $S_f = 40$, $I_{max} = 10$, $P_c = 0.8$, $P_m = 0.01$.

7.3.1 First results

In this section, we present the first results obtained for the proposed algorithm. These preliminary results have the purpose to set the best parameter values (especially the percentage of hash information to be sent, and the quantization index used for its transmission). For this reason, a set of experiments have been run on the three test video sequences of each configuration (mono/multiview). The quantity of hash information transmitted can vary

due to two parameters: the number of blocks which require a hash side information refinement (measured in %) and the quantization level (given as a QI parameter, see Table 7.1). We run several experiments in order to adopt the best configuration. We tested all of the couples (%, QI) in the set $\{2\%, 5\%\} \times \{\text{QI } 1, \text{QI } 2, \text{QI } 4, \text{QI } 6\}$. In these tests we measured the rate, r^H (due to the hash transmission), and the PSNR (the quality in dB) of the hash SI. Then we performed the turbo decoding of these obtained hash SI and measured for each couple the number of transmitted parity bits, and the quality of the final decoded WZ frame.

Figure 7.5 presents the results obtained for *foreman* (average PSNR depending on the average rate of either the hash bits or the parity bits) at a quantization step for the key frames of 31. The different couples of points represent the rate-distortion values respectively for the hash side informations and for the final turbo decoded WZ frames. Note that the final turbo decoded rate is the addition of the hash rate and the required parity bits.

What is noticeable in Figure 7.5 and was confirmed for all sequences is that the best couple is a percentage of 2% with a quantization of $QI = 4$. That means that the hash sent has to be quite precise but its rate is quite low.

Besides, in our preliminary results we have seen that the genetic algorithm, in spite of its complexity, brings a real interest compared to a simple direct hash-based fusion or inverse DCT of the received hash. Indeed, in our tests, we obtained that a candidate fusion done with the genetic algorithm could lead to an improvement of $0.2-0.5$ dB for the PSNR of the WZ estimation, compared to a simplest fusion (using the hash as reference information). In the next section, we test the performances of the proposed hash algorithm and compare the obtained rate-distortion results to the Discover reference scheme.

7.3.2 Rate-distortion results

The rate-distortion curves are shown in Figure 7.6 for the three mono-view CIF sequences and in Figure 7.7 for the multiview sequence. It can be observed that, at high bitrates, the performance of the hash-based scheme is always better than the reference. This is explained by the fact that at these rates, the hash rate is low compared to the rate of the parity information sent for turbo-decoding. On the contrary, at low bitrates, the hash rate becomes too high and the performance of the hash-based scheme is degraded for *foreman* and *mobile*. To measure the general gain, we use the Bjontegaard metric [Bjontegaard, 2001]. Though for *mobile* the average gain is almost zero, for *foreman* and *football* sequences with a less uniform motion, the gains are interesting. Indeed, the decoded quality is improved by 0.14 dB for *foreman* and 0.19 dB for *football*. Moreover, the rate reduction is around 2.7% for *foreman* and 3.0% for *football*. Improvements in the multiview setting are also acceptable. For *outdoor* sequence, the PSNR improvement is about 0.1 dB and the rate reduction is of -1.15%.

7.4 Conclusion

In this chapter, we have presented two new hash-based DVC schemes (one monoview and one multiview) which present two novelties. Firstly, the hash information selection is performed at the decoder (and not at the encoder as in the previous hash-based schemes)

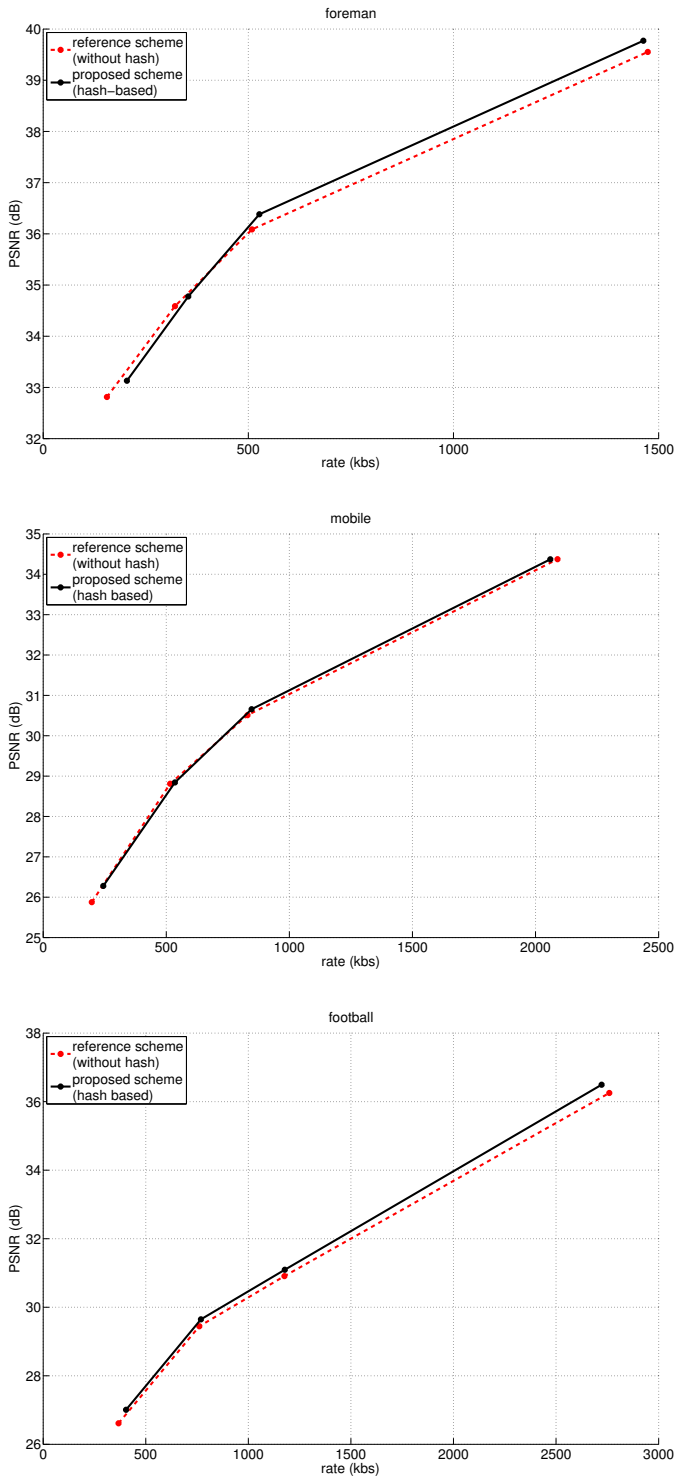


Figure 7.6: RD performances for three CIF test sequences. In dashed red lines, the DISCOVER reference scheme, in plain black, the proposed adaptative hash-based algorithm.

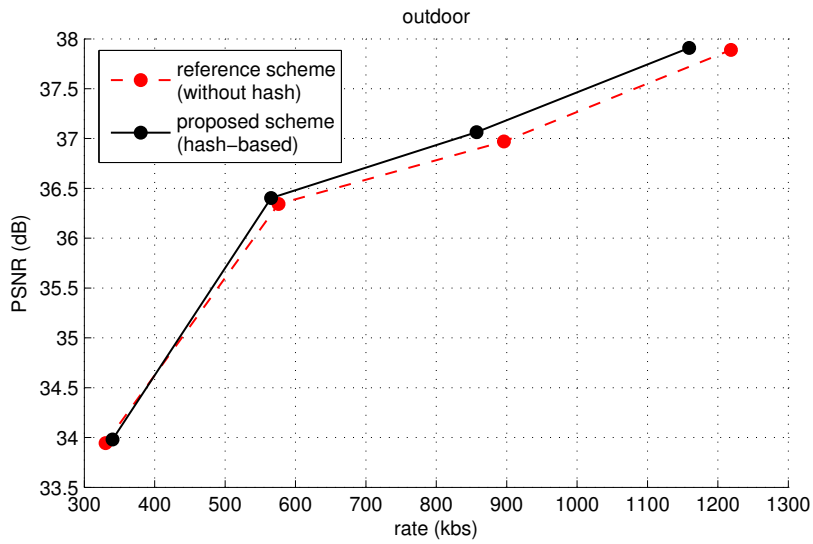


Figure 7.7: RD performances for *outdoor* multiview test sequences (512×384). In dashed red lines, the Discover reference scheme, in plain black, the proposed adaptative hash-based algorithm.

and thus uses the true side information. This information is more pertinent than the knowledge of the true WZ frames, as it is the case when the hash selection is performed at the encoder. Moreover, we propose to use a genetic based fusion algorithm which aims at merging several efficient temporal and inter-view interpolations. The experimental results confirmed that the proposed approach can lead to interesting improvements.

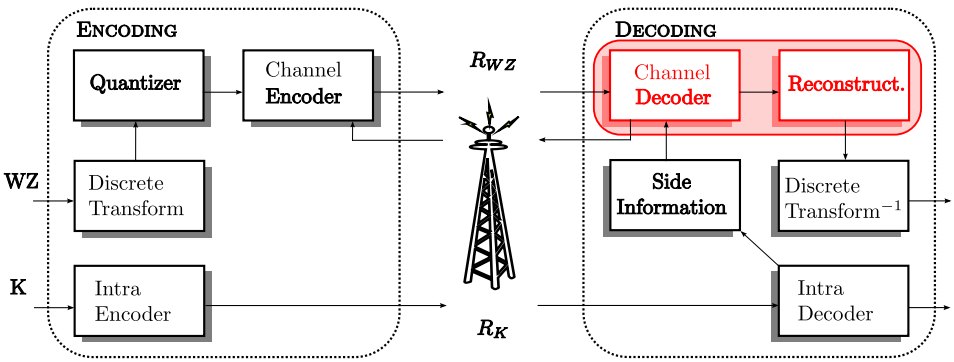
However, the proposed hash-based scheme has two drawbacks. These ones are already existing classical disadvantages of DVC, but they are deepened in our architecture. Firstly, our scheme accentuates the need of a return loop, since it performs the hash selection at the decoder. Moreover, the decoding complexity is sensibly increased by all of the GA candidates, especially in the multiview setting.

Acknowledgment

This work was partly supported by a research grant from the Lebanese National Council for Scientific Research (LNCSR) and was realized within the Franco-Lebanese CEDRE (08 SCI F2 / L1) program.

The monoview algorithm presented in this chapter was published in:

- T. Maugey, C. Yaacoub, J. Farah, M. Cagnazzo, and B. Pesquet-Popescu, "Side information enhancement using an adaptive hash-based genetic algorithm in a wyner-ziv context," in *Int. Workshop on Multimedia Sig. Proc.*, Saint-Malo, France, Oct. 2010.



Part III

Zoom on Wyner Ziv decoding

"A better understanding of what happens at the WZ decoder."

Chapter 8

Correlation noise estimation at the Slepian-Wolf decoder

The most popular channel codes used in distributed video coding are the turbocodes or the LDPC. Both of them require an estimation of the a priori probabilty of the variable X (to decode) and its side information Y. The precision of this estimation has a strong impact on the error correction efficiency, and thus on the quantity of parity information required.

This a priori probability, $p_{X|Y}(X)$, is also called correlation noise. *Its estimation consists in modelling the error distribution $X - Y$ with a probability density function (pdf) $f_{X|Y}(X)$. The Slepian-Wolf decoder performs the integration of this pdf to compute the* a priori *probabilities used for error correction.*

In this chapter, we first perform a detailed review of the existing correlation noise estimations techniques (Section 8.1), and then we will propose to use the Generalized Gaussian model intead of the commonly adopted Laplacian one (Section 8.2). Finally, based on the observation that a better fitted distribution does not necessarily improve the decoding efficiency, we propose a more complete study in Section 8.3.

Contents

8.1 State-of-the-art: existing models 186
8.1.1 Pixel domain 186
8.1.2 Transform domain 190
8.1.3 Performance evaluation 192
8.2 Proposed model: Generalized Gaussian model 193
8.2.1 Definition and parameter estimation 193
8.2.2 Approach validation 195
8.2.3 Experimental results 197
8.3 A more complete study 199
8.3.1 Motivations 199
8.3.2 Experiments and results 200
8.3.3 Conclusion 205

The purpose of several works on correlation noise for distributed video coding was to estimate a faithful disribution, and almost all of them are based on a Laplacian model. The problem is that the frame X is not available at the decoder, and thus the distribution of error $X-Y$ cannot be directly estimated. Two approaches are considered in the literature:

- **Offline** - In this configuration, the true error $X-Y$ is used for correlation noise estimation. It is unrealistic since this estimation is performed at the decoder, but it gives interesting results of "ideal" estimation (like an oracle). The offline configuration is represented in red in Figure 8.1.
- **Online** - In this approach, the error $X-Y$ is estimated by another residual. This residual is usually [Girod *et al.*, 2005; Artigas *et al.*, 2007a] $\frac{Y_1-Y_2}{2}$, where Y_1 and Y_2 are two versions of the side information, like the two motion compensated reference frames. This is shown in the green part of Figure 8.1.

8.1 State-of-the-art: existing models

All of the existing solutions use a Laplacian model for the correlation noise estimation. The Laplacian distribution is given by

$$\forall x \in \mathbb{R}, \qquad f_{lap}(x) = \frac{1}{2\alpha} e^{-\frac{|x|}{\alpha}} \qquad \text{where} \quad \alpha \in \mathbb{R}^{+*}.$$

This model is popular since it roughly corresponds to the true error distribution in practice (we will see in Section 8.2 that it often happens that the model is limited and does not propose a good and fine description of the distribution). Another reason of its utilization is its simplicity. Indeed, only one coefficient, α has to be estimated and it is a memory less model.
It is however obvious that the error is not stationary (in time and space), because the motion activity differs in different regions of the image and at different instants in the sequence. The α parameter can thus be estimated at different levels of precision, while dealing with the compromise between time or space precision (α estimated with a few samples) and the statistic precision (α estimated with a lot of samples).
The literature proposes several ways of estimating α. They differ from the level of precision (sequence, frame, band, macroblock, coefficient, pixel) and from domain (transformed or pixel). The following is a description of some of these methods. One of the most relevant work is the one by of Brites et Pereira [Brites, Pereira, 2008] who give a detailed comparison of each level of precision. Most of the methods described in the following review of literature come from this work. In the following section, the estimation error variance is denoted by σ^2 when it is calculated with the true original frame (offline) and $\hat{\sigma}^2$ when it is calculated with the residual (online setting).

8.1.1 Pixel domain

For pixel domain distributed video coding schemes, the channel encoding/decoding of the WZ frames is performed in the pixel space, and then, the correlation noise is also estimated as a pixel estimation error. All of the different existing levels of precision are represented in Figure 8.2. The major works in correlation noise estimation in pixel-domain have again been proposed by Brites et al. [Brites *et al.*, 2006d; Brites *et al.*, 2006c].

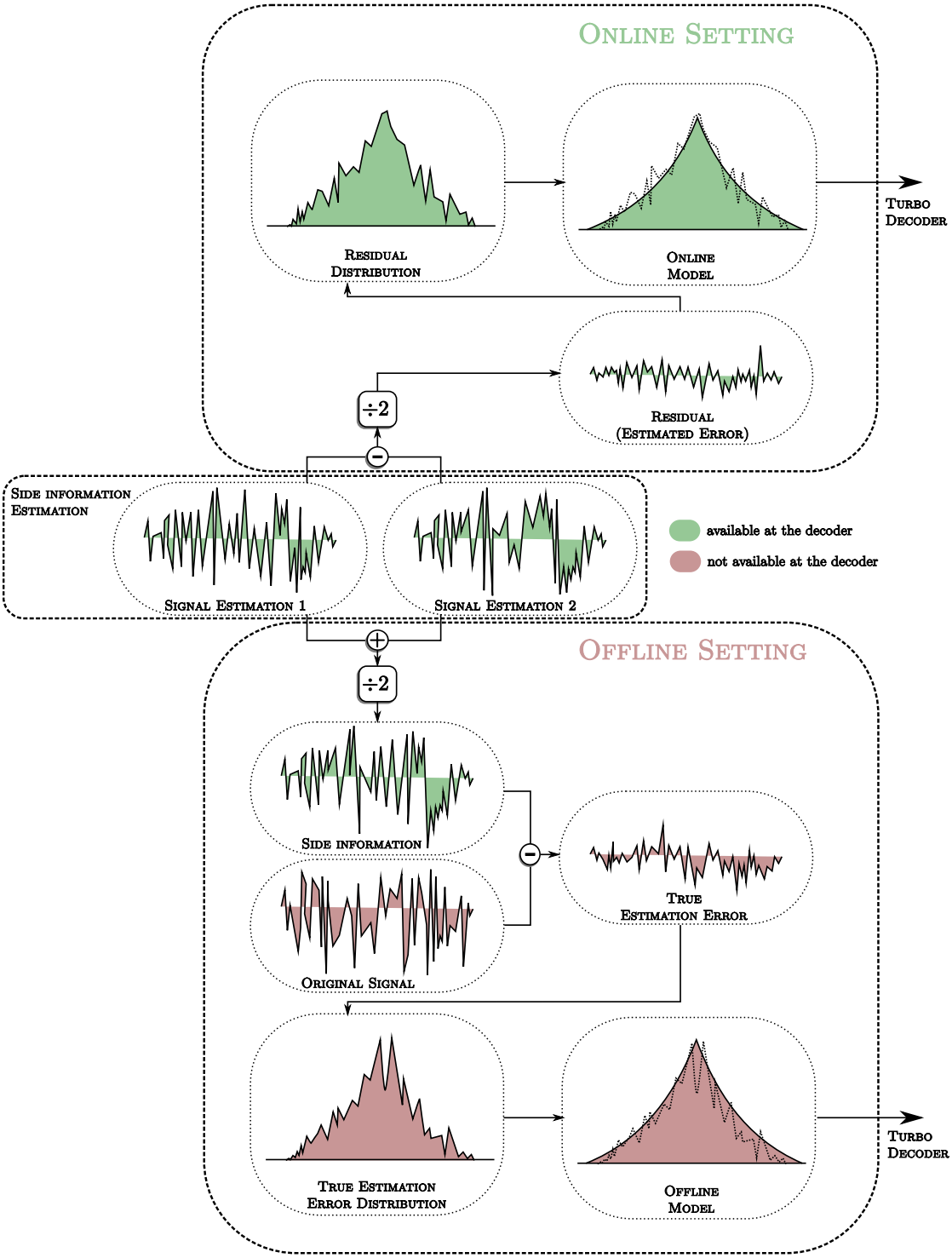


Figure 8.1: Online and offline general description for correlation noise estimation at the Wyner-Ziv decoder

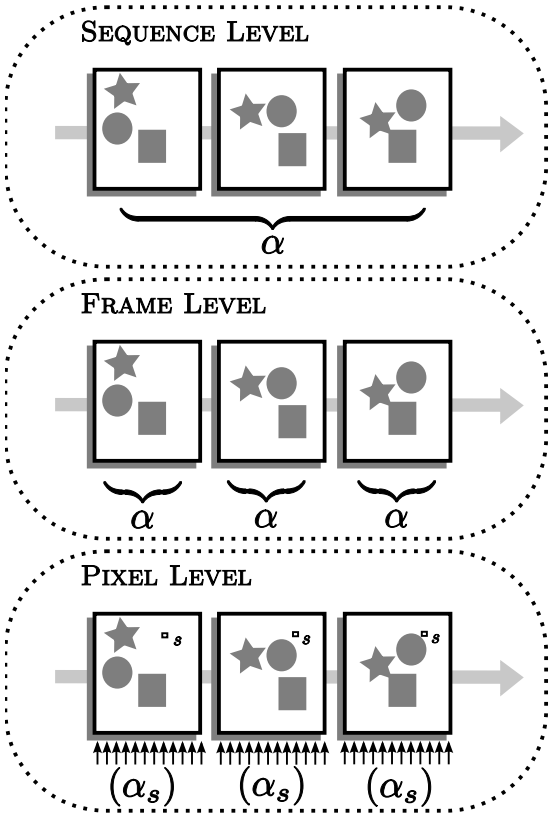


Figure 8.2: Existing levels of precision for α parameter estimation in pixel domain.

8.1.1.1 Sequence level

The sequence level parameter estimation consists in setting one value of α for the whole sequence. In [Brites *et al.*, 2006d], Brites et al estimate this parameter offline. They compute the average variance of the error, $\sigma^2_{sequence}$ along the sequence and deduce $\alpha^{off}_{sequence}$ from the well-known relation

$$\alpha^{off}_{sequence} = \frac{\sigma_{sequence}}{\sqrt{2}}.$$

In this case, we have a very coarse approximation of the correlation between the Wyner-Ziv frame and its side information, because the assumption of stationarity along the sequence is not often verified. Moreover this sequence level approach is not proposed with online α estimation.

8.1.1.2 Frame Level

The frame level precision starts to overcome the non-stationarity of the noise correlation. Indeed, instead of calculating the α for all the sequence, it is evaluated for each frame. The process is however similar to the one of the sequence level. Indeed, for the offline setting, the variance of the estimation error, σ^2_{frame}, is calculated and used for deducing the corresponding α_{frame}:

$$\alpha^{off}_{frame} = \frac{\sigma_{frame}}{\sqrt{2}}.$$

For the online setting, instead of the true error variance, the decoder calculates the variance of the residual, $\hat{\sigma}^2_{frame}$, (the difference between the two motion compensated frames divided by two):

$$\alpha^{on}_{frame} = \frac{\hat{\sigma}_{frame}}{\sqrt{2}}.$$

For distributed video coding in a multicamera configuration (with hybrid or symmetric frame type repartition, see Section 3.1.1 for more details), Avudainayagam et al. adopt a similar approach in [Avudainayagam *et al.*, 2008] but take into account the 4 reference frames (instead of 2).
Deligiannis et al. in [Deligiannis *et al.*, 2009] also proposed a Laplacian frame level noise correlation estimation, but their Laplacian model is a little more sophisticated because it takes into account the variance of the side information.

8.1.1.3 Block level

The temporal non-stationarity (along the sequence) is resolved by frame level precision. On the other hand, it is accepted that the correlation noise is also spatially non-stationary. Since some regions of the image are badly estimated (occlusions, rapid motion, etc.) and other are well estimated. That is why Brites et al. propose to be more precise and decide to evaluate α for each 8×8 macro-block. The evaluation method slightly differs from the other level. Indeed the estimation error variance is calculated block by block but now, this value is taken into account only if the block variance is greater than 1 for offline setting[1]

[1] to avoid a zero or too little value

and greater than the frame variance:

$$\alpha_{block}^{off} = \max\left\{\frac{\sigma_{block}}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}$$
$$\alpha_{block}^{on} = \max\left\{\frac{\hat{\sigma}_{block}}{\sqrt{2}}, \alpha_{frame}^{on}\right\}$$

This approach reflects the choice, in online setting, to overestimate the noise correlation, *i.e.*,to set the lowest α_{block} at α_{frame}. In fact, the behind assumption is that the correlation is stationary except where the side information diverges.

8.1.1.4 Pixel Level

Because the block stationarity is still a too strong assumption, Brites et al. propose to refine once more the α estimation by adopting a similar approach for pixel estimation. In other words, the block evaluation is assumed to be stationary except when the square error, e_{pixel}^2 is lower than the block error variance for online setting (and greater than 1 in offline configuration). Moreover, for online estimation, the technique also takes into account the quantity D_{block} which is the square of the difference between the average of residual on the block and on the entire frame.

$$\alpha_{pixel}^{off} = \begin{cases} \frac{1}{\sqrt{2}} & \text{if} \quad \sigma_{block}^2 \leq 1 \\ \frac{|e_{pixel}|}{\sqrt{2}} & \text{if} \quad \sigma_{block}^2 > 1 \end{cases}$$

$$\alpha_{pixel}^{on} = \begin{cases} \alpha_{frame}^{on} & \text{if} \quad \hat{\sigma}_{block}^2 \leq \hat{\sigma}_{frame}^2 \\ \alpha_{block}^{on} & \text{if} \quad \hat{\sigma}_{block}^2 > \hat{\sigma}_{frame}^2 \text{ and } D_{block} \leq \hat{\sigma}_{frame}^2 \\ \alpha_{block}^{on} & \text{if} \quad \hat{\sigma}_{block}^2 > \hat{\sigma}_{frame}^2 \text{ and } D_{block} > \hat{\sigma}_{frame}^2 \text{ and } \hat{e}_{pixel}^2 \leq \hat{\sigma}_{block}^2 \\ \frac{|\hat{e}_{pixel}|}{\sqrt{2}} & \text{if} \quad \hat{\sigma}_{block}^2 > \hat{\sigma}_{frame}^2 \text{ and } D_{block} > \hat{\sigma}_{frame}^2 \text{ and } \hat{e}_{pixel}^2 > \hat{\sigma}_{pixel}^2 \end{cases}$$

The Brites method is more advanced than the one of Qing et al. [Qing *et al.*, 2007] which does not perform such thresholding considerations and then sometimes diverges.

8.1.2 Transform domain

Transform domain noise correlation also consists in estimating an error variance, σ, and deduces the value of α by the same relation used in the spatial domain. Nevertheless, the error variance is estimated in the transform domain (commonly 4×4 DCT) and must be different for each of the 16 bands. The "transform domain" estimation is thereby executed for each band, and as before, the existing methods vary from their level of precision (Figure 8.3). In the following, *band* denotes the band index.

8.1.2.1 Sequence level

As in the pixel domain configuration, the estimation error variance along the sequence is estimated but this time, $\sigma_{sequence}^2(band)$. The α calculation is then:

$$\alpha_{sequence}^{off}(band) = \frac{\sigma_{sequence}(band)}{\sqrt{2}}.$$

It does not exist an equivalent online estimation, whereas it would not be difficult to extend the previous equation to online settings, but the inprecision due to temporal and spatial

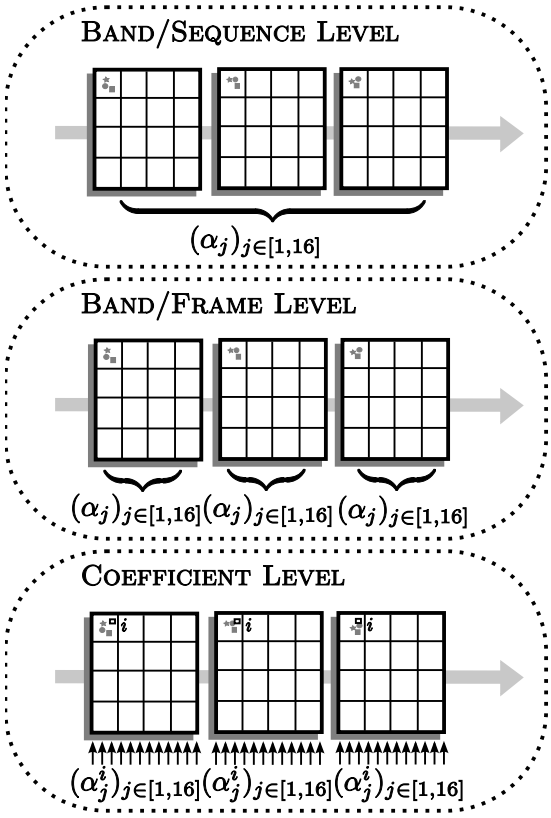


Figure 8.3: Existing levels of precision for α parameter estimation in DCT domain.

stationarity assumption, added to the inprecision of the residual, would lead to a too coarse estimation of α.

8.1.2.2 Frame Level

The frame level precision α estimation is obtained by once again calculating the estimation error variance for each band and for each frame, and thus having

$$\alpha^{off}_{frame}(band) = \frac{\sigma_{frame}(band)}{\sqrt{2}}$$
$$\alpha^{on}_{frame}(band) = \frac{\hat{\sigma}_{frame}(band)}{\sqrt{2}}.$$

In [Slowack *et al.*, 2009], Slowack et al propose to take into account the quantization noise in the online setting α estimation for a frame level precision. Indeed, the residual is obtained by calculating the difference between the two motion compensated reference frames which are quantized. The method is quite efficient especially when the quantization is very coarse.

8.1.2.3 Coefficient level

The estimation of α at coefficient level proposed by Brites uses the quantity $|t(band, coefficient)$ (respectively $\hat{t}(band, coeff)|$) which is the 4×4 DCT transform of the image error (respectively of the residual) for an offline (respectively online) setting. The $\alpha_{coeff}(band)$ is given by

$$\alpha^{off}_{coeff}(band) = \max\left(\frac{1}{\sqrt{2}}, \frac{|t(band, coeff)|}{\sqrt{2}}\right)$$
$$\alpha^{on}_{coeff}(band) = \max\left(\alpha^{on}_{frame}(band), \frac{|\hat{t}(band, coeff) - \mu(band)|}{\sqrt{2}}\right)$$

where $\mu(band)$ is the mean of $\hat{t}(band, coeff)$ with respect to $coeff$.

Several works in the literature adopt the coefficient level precision. They propose alternative approaches but retain the same hypotheses: a Laplacian model whose parameter is estimated for every coefficient of each band. Dalai and Pereira [Dalai *et al.*, 2006] estimate α as a function of global frame statistics (error variance per band) and also based on the confidence the decoder can have in the side information (which is estimated by the residual). Later, Esmaili et al. [Esmaili, Cosman, 2009] determine a set of several modes (of possible α values), then the decoder guess coefficient by coefficient the most appropriate mode (the modes correspond in fact to different statistics in the scene as background, rapid motion object, etc.). The ideas of coefficient classification is also adopted in Huang and Forchhammer work [Huang, Forchhammer, 2009].

8.1.3 Performance evaluation

All of these works demonstrate that refining the correlation noise model sensibly improves the performances. However, the gains are quite limited in some cases, like the results by Brites have shown. Indeed, switching from a frame level to a pixel level precision for online setting reduces the required rate by 6%, which is acceptable, but only by 0.5 in some

situations depending on the sequence and the bitrate. Rate gains can be greater in offline settings but they do not transpose in the same proportion in the RD gains. Nevertheless, they show what can be the maximum evolution gap and encourage to continue the correlation noise model refinement, even though the gains are each time limited.

The performances also show that a coefficient level precision does not bring acceptable gains especially in online settings. Indeed, it is useless to be very precise with a residual which is already a limited estimation. That is why, in the following, we adopt a DCT frame level precision, while still comparing our proposition to coefficient level configuration, which is actually the reference in correlation noise estimation.

8.2 Proposed model: Generalized Gaussian model

As it can be seen in Figure 8.4 the Laplacian model does not always fit the error distribution in distributed video coding and a refinement of the model seem to be justified. We thus propose here to use the more general *Generalized Gaussian (GG)* model which is potentially enable to better fit the true distribution.

The material in this section was published in

- T. Maugey, J. Gauthier, B. Pesquet-Popescu, and C. Guillemot, "Using an exponential power model for wyner-ziv video coding," in *Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, Dallas, Texas, USA, Mar 2010.
- J. Gauthier, T. Maugey, B. Pesquet-Popescu, and C. Guillemot, "Amélioration du modèle statistique de bruit pour le codage vidéo distribué," in *Proc. GRETSI*, Dijon, France, Sep. 2009.

8.2.1 Definition and parameter estimation

The pdf of Generalized Gaussian (or Exponential Power Distribution, $\mathcal{EPD}$) with zero mean and parameters $\alpha \in \mathbb{R}_+^*$ and $\beta \in \mathbb{R}_+^*$ reads

$$f_{gg}(x) = \frac{\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)} e^{-\left(\frac{|x|}{\alpha}\right)^{\beta}},$$

where $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ is the classical "gamma" function. Several methods are available to compute the parameters of an $\mathcal{EPD}$, among them the maximum likelihood estimation and the moment estimation. In this section we give some details about these two classical estimation methods, which will then be compared in the DVC framework.

8.2.1.1 Moment estimation

A first idea to estimate (α, β) is to compute the moments of order 2 and 4, leading to: $\mu_2 = \alpha^2 \frac{\Gamma\left(\frac{3}{\beta}\right)}{\Gamma\left(\frac{1}{\beta}\right)}$ and $\mu_4 = \alpha^4 \frac{\Gamma\left(\frac{5}{\beta}\right)}{\Gamma\left(\frac{1}{\beta}\right)}$. Combining these two formulas, the kurtosis κ can be

expressed as a function of β: $\kappa = \frac{\mu_4}{\mu_2^2} = \frac{\Gamma\left(\frac{5}{\beta}\right)\Gamma\left(\frac{1}{\beta}\right)}{\Gamma\left(\frac{3}{\beta}\right)^2} = g(\beta)$. Finally, the parameters α and β can be estimated by:

$$\widehat{\beta} = g^{-1}(\kappa), \quad \widehat{\alpha} = \sqrt{\frac{\Gamma\left(\frac{1}{\widehat{\beta}}\right)}{\Gamma\left(\frac{3}{\widehat{\beta}}\right)}\mu_2}. \tag{8.1}$$

This method thus relies on the estimation of the variance and kurtosis of the observed samples, and on the inversion of the function $g : \mathbb{R}_+^* \to \mathbb{R}_+^*$. This function being strictly decreasing, it is possible to compute a unique $g^{-1}(\kappa)$ for all $\kappa \in \mathbb{R}_+^*$.

8.2.1.2 Maximum likelihood estimation

Our goal in this section is again to find an estimation of α and β given a set of independent observations $\boldsymbol{\xi} = (\xi_i)_{1 \leq i \leq N}$. The pdf of the joint distribution reads

$$F_{\alpha,\beta}(\boldsymbol{\xi}) = \left(\frac{\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)}\right)^N e^{-\sum_{i=1}^N \left(\frac{|\xi_i|}{\alpha}\right)^\beta}.$$

The anti log-likelihood can be expressed as:

$$p(\alpha, \beta|\boldsymbol{\xi}) = -\ln(F_{\alpha,\beta}(\boldsymbol{\xi})) = \sum_{i=1}^N \left(\frac{|\xi_i|}{\alpha}\right)^\beta + N\left(\ln(\alpha) - \ln\left(\frac{\beta}{2\Gamma\left(\frac{1}{\beta}\right)}\right)\right). \tag{8.2}$$

To minimize the anti log-likelihood, which is tantamount to maximizing the likelihood, we first differentiate $p(\alpha, \beta|\boldsymbol{\xi})$ with respect to α:

$$\frac{\partial p(\alpha, \beta|\boldsymbol{\xi})}{\partial \alpha} = -\frac{\beta}{\alpha^{\beta+1}} \sum_{i=1}^N |\xi_i|^\beta + \frac{N}{\alpha}.$$

Looking for the zeros of this partial derivative we get α_{min} as a function of β:

$$\alpha_{min} = \left(\frac{\beta}{N}\sum_{i=1}^N |\xi_i|^\beta\right)^{\frac{1}{\beta}}. \tag{8.3}$$

Combining (8.2) and (8.3), we obtain:

$$p(\widehat{\alpha}, \beta|\boldsymbol{\xi}) = \frac{1}{\beta} - \ln\left(\frac{\beta}{\Gamma\left(\frac{1}{\beta}\right)}\right) + \frac{1}{\beta}\ln\left(\frac{\beta}{N}\sum_{i=1}^N |\xi_i|^\beta\right) = h(\beta).$$

Finally, we compute $\widehat{\beta}$ as the argmin of h and we get $\widehat{\alpha}$ by replacing β by $\widehat{\beta}$ in (8.3).

8.2.1.3 Comparison

Both methods are tested against different generated $\mathcal{EPD}$ vectors with given parameters (α, β). The size of the observation vector was set to 6336 to match the number of coefficients

in a DCT subband of a CIF video frame. For the different chosen parameters, the computed values were very close to the real ones with both methods (the mean square error being less than 10^{-3}). In Table 8.1, the variances of the estimated parameters over 100 observation vectors are reported. While with both methods and for all the tested combinations of parameters the variances on the estimated parameters are small, the moment method presents a slightly higher deviation. It should finally be noted that the complexity of the moment method is sensibly lower than that of the maximum likelihood method (with our MATLAB implementation, the moment method is almost 80 times faster).

(α, β)	Moment	ML
$(1.5, 1)$	$(0.116, 0.048)$	$(0.052, 0.022)$
$(3, 2)$	$(0.054, 0.065)$	$(0.036, 0.032)$
$(1.25, 1.5)$	$(0.033, 0.047)$	$(0.028, 0.037)$

Table 8.1: Standard deviations over 100 observed vectors of $(\widehat{\alpha}, \widehat{\beta})$ for different values of (α, β) (corresponding to Laplacian, Gaussian and Generalized Gaussian distributions).

8.2.2 Approach validation

Before testing the benefits of using a more precise estimation, we study whether the decoding performances are improved by using a model which better fits the actual noise distribution.
For a band b, the error discretly lies between a minimum value, min, and a maximum value, max. In this range, a model is estimated at the decoder, the obtained function is denoted by f (the associated discrete probability mass function, *i.e.*,the discrete value multiplied by the length of the bin, is denoted by f^*). Let H_b be the distribution of the error (*i.e.*,the histogram of error values). To evaluate the discrepancy between H_b and f, many classical measures can be considered. We have chosen the following family of functions:

$$d^a(f, H_b) = \sum_{n=min}^{max} |f^*(n) - H_b(n)|^a ,$$

where $a \in \mathbb{R}_+^*$. For each band b of a given frame, two models are estimated, f_1 and f_2. The decoding of this band is performed and the obtained rate is denoted by r_b^1 if f_1 has been used for calculating the *a priori* probabilities for the turbo decoding (respectively r_b^2 if f_2 has been used). We recall that this rate corresponds to the number of bits required to reach a bit error probability lower than 10^{-3}. Let a be in $\mathbb{R}_+^*$ and let us introduce the following Hypothesis, Hyp:

$$\begin{gathered}\text{For each band, } \forall (i,j) \in [1,2]^2, i \neq j, \\ d^a(f_i, H_b) \leq d^a(f_j, H_b) \Leftrightarrow r_b^i \leq r_b^j\end{gathered}$$

Minimizing the distance between H_b and f, *i.e.*,improving the error distribution model, is justified only if Hyp is true. For four CIF test sequences, we test for every band of every frame if Hyp is verified. In the experiments, f_1 and f_2 correspond to respectively a Laplacian and an $\mathcal{EPD}$ distributions. The obtained results are presented in Tab. 8.2 for

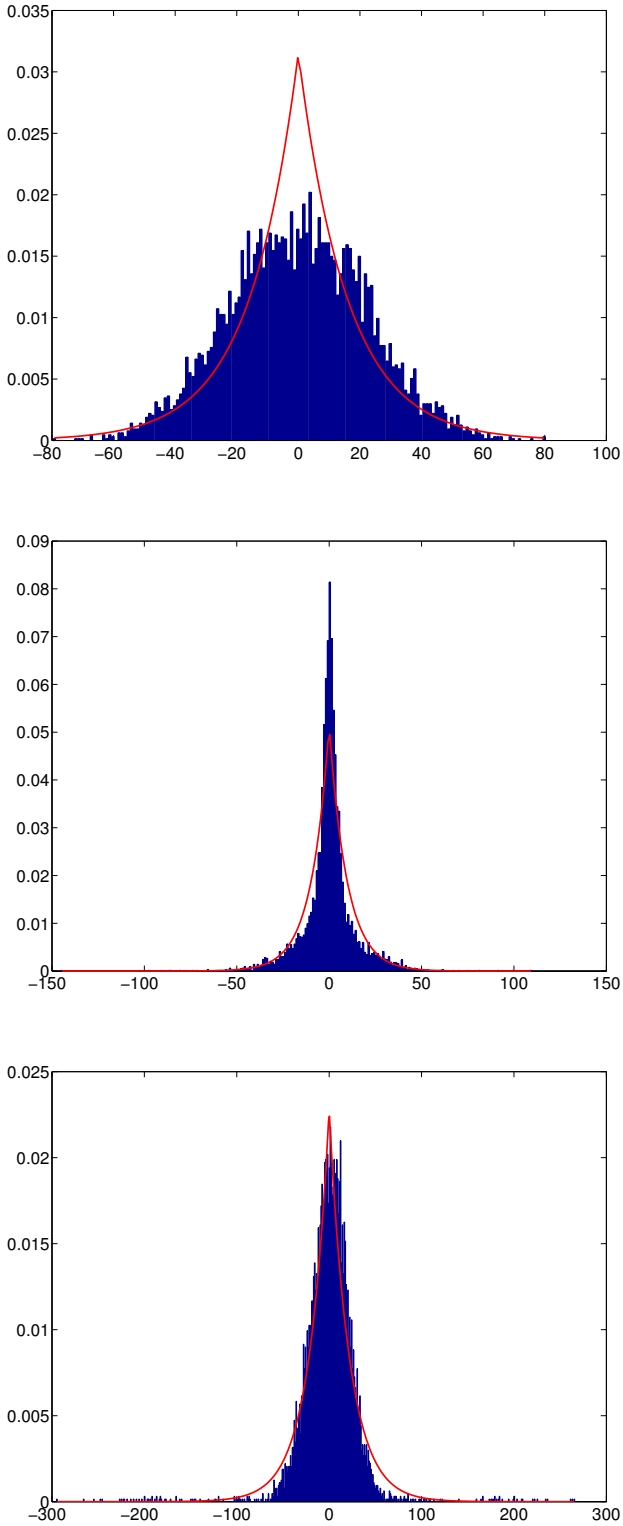


Figure 8.4: Examples of error distributions and the best fitted Laplacian model, for different bands and different sequences.

	d^2	d^1	$d^{\frac{1}{2}}$	$d^{\frac{1}{3}}$
waterfall	97	97	97	97
foreman	94	91	91	97
football	82	94	94	82
mobile	94	85	88	88

Table 8.2: % of measures where Hyp was verified.

$a \in \{2, 1, \frac{1}{2}, \frac{1}{3}\}$, corresponding to the most representative values among the experimental set.
The obtained statistics show that there is a strong correlation between the distances d^a and the measured rates. In other words, attempting to fit well the histogram is justified by the fact that in this way it is likely to improve the performances. Based on this idea, in the next section we test the performances of the $\mathcal{EPD}$ distribution.

8.2.3 Experimental results

In the previous section we proved that fitting well to the error distribution can improve the coding performances. In this section we test the coding efficiency of using an $\mathcal{EPD}$ instead of the classical Laplacian model employed in the literature.

8.2.3.1 Experimental setting

The presented experimental results were obtained with the DVC scheme described in the introduction. Tests were run on two CIF video sequences: "City" and "Football" (352×288, 30Hz) and one QCIF sequence: Foreman (176×144, 15Hz). The 100 first frames (50 KFs, and 50 WZFs) of each sequence are coded, and for each coding configuration, the average rate (in kbs) has been measured. To cover a wide range of rates, the methods have been tested at four quantization levels (Q-Index for the WZFs | Q-Step for H.264 intra coding of the KFs) chosen as follows: 1|42, 4|34, 6|31 and 8|28.
Tests are run both for the Laplacian and the $\mathcal{EPD}$ models, with the online and offline coefficient estimation modes. For the $\mathcal{EPD}$ model, the maximum likelihood (ML) and moment (Mom) estimation methods are both employed for "on/offline" parameter prediction. Results are shown in Tab. 8.3, presenting the average rate gain (in %). These gains are estimated with the Bjontegaard metric [Bjontegaard, 2001]. Additional results are shown in Tabs. 8.4 and 8.5, presenting the bitrates obtained by different methods for the four quantization levels on the CIF Football sequence and QCIF Foreman sequence. Finally, Fig. 8.5 presents the RD results of the different models for the CIF Football sequence. The following notations are used in these tables: "Lapl" stands for Laplacian method and "On", resp. "Off" mean online and offline estimation modes.

8.2.3.2 Comparison in the offline setting

We first compare the results of the different methods in the offline mode. The corresponding results on the test sequences can be read from the first two lines of Tab. 8.3, 8.4 and 8.5 and from the red plots of Fig. 8.5. We see that on both videos the $\mathcal{EPD}$ model (in ML or Mom case) needs a smaller bitrate than the Laplacian model, with average bitrate gains up

to 3.73% for Football (CIF) and 1.78% for Foreman (QCIF). At high bitrate for these two sequences, the transmission rate can be reduced by 194kbs with a CIF video and 44kbs with a QCIF sequence. Another interesting conclusion is that the maximum likelihood estimation performs systematically better than the moment method.

8.2.3.3 Comparison in the online scenario

A second comparison is performed in the online mode. The results in this case are reported in the third and fourth lines of Tab. 8.3. Black plots in Fig. 8.5 also present the online mode results for the Football sequence. Once again, the $\mathcal{EPD}$ model outperforms the Laplacian model. Yet, it is interesting to note that unlike the offline setting, the moment method yields better results than the ML, meaning that the moment estimation method seems more robust. The bitrate gain reaches 4.3% for the Football (CIF) sequence and 1.88% for the Foreman video (QCIF). In Tabs. 8.4 and 8.5 we see that in the online mode the $\mathcal{EPD}$ method reduces the transmission rate by 128kbs for a CIF video and by 46kbs for a QCIF sequence when compared with the Laplacian method. This realistic scheme also outperforms H.264 intra coding (7% of rate saving, and 0.35dB of quality improvement for the Football sequence).

8.2.3.4 Comparison between the offline and online settings

Finally, we compare the results obtained in the offline and online settings. Considering the fifth and sixth lines of Tab. 8.3, it is worth noting that the loss incurred by switching from offline to online is slightly higher with the Laplacian model.
The last considered case is the comparison between Laplacian offline and $\mathcal{EPD}$ online, with results reported in the last line of Tab. 8.3 and in Fig. 8.5. It is interesting to note that the online results obtained with $\mathcal{EPD}$ are better than the offline results with the Laplacian model for the Football and Foreman sequence. In other words, it means that the $\mathcal{EPD}$ model with parameters computed without knowledge of the original WZ performs better than the Laplacian model with parameters estimated with this knowledge. For the City sequence, these rates are close (0.44%) when considering the whole bitrate range. Note that for this last sequence with high bitrates (1600kbs to 4000kbs), we observe that the $\mathcal{EPD}$ online performs slightly better (0.75% gain in bitrate) than the Laplacian offline.

Method 1	Method 2	City	Football	Foreman
Lapl Off	$\mathcal{EPD}$ Off ML	-0.96	-3.73	-1.78
Lapl Off	$\mathcal{EPD}$ Off Mom	1.21	-3.61	-1.52
Lapl On	$\mathcal{EPD}$ On ML	0.36	-3.29	-0.90
Lapl On	$\mathcal{EPD}$ On Mom	-1.3	-4.30	-1.88
Lapl Off	Lapl On	1.73	2.67	1.53
$\mathcal{EPD}$ Off ML	$\mathcal{EPD}$ On Mom	1.4	2.10	1.39
Lapl Off	$\mathcal{EPD}$ On Mom	0.44	-1.64	-0.38

Table 8.3: Rate gains (%) by method 2 over method 1 on City, Football (CIF, 30Hz) and Foreman (QCIF, 15Hz) sequences.

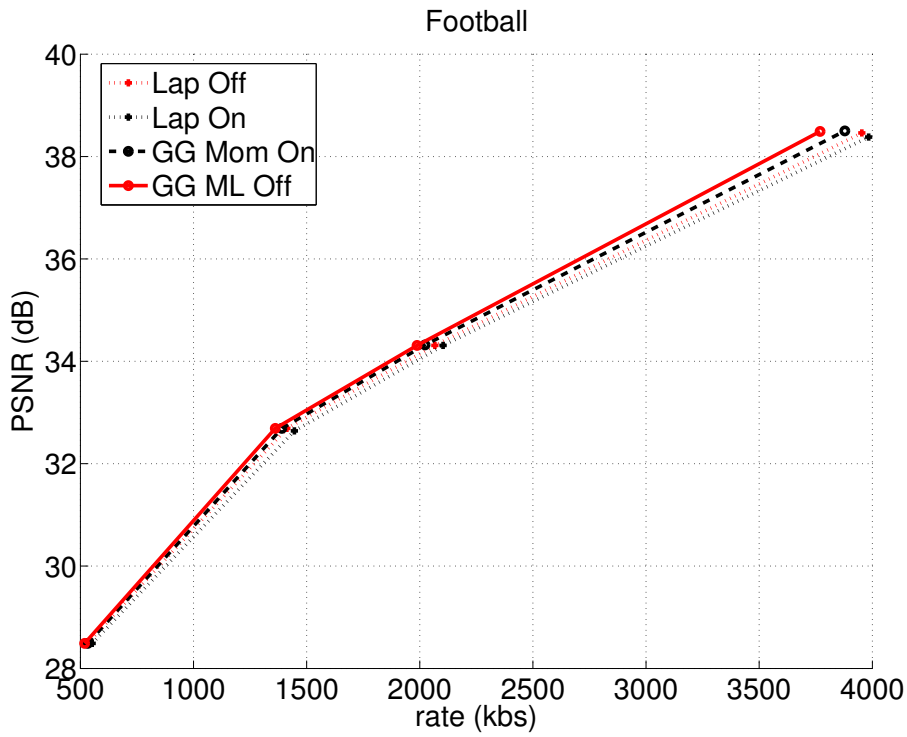


Figure 8.5: Rate-distortion performance for *football* sequence, CIF, 100 frames, 30 fps.

8.2.3.5 Discussion

Knowing that a better fitted distribution enables an improvement of the RD permormances, the purpose of these tests is to measure the reliability of the $\mathcal{EPD}$ model. Experimental results have shown that the $\mathcal{EPD}$ model is finer than the Laplacian one, yielding bitrate improvements on the considered test sequences. Improvements may of course vary from one video to another depending on how close the residual distribution is to a Laplacian one. We also want to emphasize that the gains obtained in this paper can be compared to those offered by other works involving refinements of the noise model [Brites, Pereira, 2008; Brites *et al.*, 2006d].
Moreover, another purpose of this work was to propose a realistic model, in the sense that it does not need the knowledge of the original WZ frame. This is precisely what is shown in Sec. 8.2.3.3 and 8.2.3.4. Indeed, we proposed an efficient online solution, which even outperforms the offline standard technique in some cases.

8.3 A more complete study

8.3.1 Motivations

Results presented in the previous section were satisfying for a set of sequences (of different spatial and temporal resolutions), and thus proved that refining the model by using a more general distribution could be an amelioration of the system.
However, while testing the GG efficiency, we obtained some suprising results. Indeed, in

PSNR (in dB)	28.49	32.64	34.31	38.38
Lapl OFF	531	1402	2066	3916
$\mathcal{EPD}$ OFF MV	519	1351	1988	3722
Δrate (kbs)	**−12**	**−51**	**−78**	**−194**
Lapl ON	552	1448	2103	3953
$\mathcal{EPD}$ ON Mom	532	1380	2019	3825
Δrate (kbs)	**−20**	**−68**	**−84**	**−128**

Table 8.4: Rate results (kbs) on the Football sequence (CIF, 30Hz) for different values of average PSNR.

PSNR (in dB)	31.36	34.4	36.44	39.94
Lapl OFF	225	424	624	1055
$\mathcal{EPD}$ OFF MV	224	421	611	1009
Δrate (kbs)	**−1**	**−3**	**−13**	**−44**
Lapl ON	227	432	632	1080
$\mathcal{EPD}$ ON Mom	226	425	622	1034
Δrate (kbs)	**−1**	**−7**	**−10**	**−46**

Table 8.5: Rate results (kbs) on the Foreman sequence (QCIF, 15Hz) for different values of average PSNR.

some situations (an example in Figure 8.6), the GG distribution which fits the true error distribution much better than the Laplacian in the offline setting, leads to the same rate for a equivalent decoded quality. In other words, in some cases, a better fitted distribution does not lead to a compression improvement.

Based on this observation, we aim at understanding what does a "good fitted" distribution mean. In other words, we need to study under which metric (MSE or another) the model has to fit the true error. Experimental principles and their results are presented in next section.

8.3.2 Experiments and results

8.3.2.1 Experiments setting and results

We denote the histogram of the true error by $h(x)$ where x is a possible error value. Let $f_{\alpha,\beta}$ be the pdf of a proposed GG model whose parameters are (α, β). In the following we aim at determining an appropriate distance metric d for measuring the difference between the histogram and the model, $d(h, f_{\alpha,\beta})$. A distance, d would be appropriate if when $d(h, f_{\alpha,\beta})$ is minimum, the turbodecoding with $f_{\alpha,\beta}$ model is optimal. The distance is in fact computed with the discrete version of $f_{\alpha,\beta}$ denoted by $f^*_{\alpha,\beta}$, whose values correspond to the values of $f_{\alpha,\beta}$ multiplied by the bin length.

The most obvious distance is the SSD distance:

$$d_{SSD}(h, f_{\alpha,\beta}) = \sum_x \Big(h(x) - f^*_{\alpha,\beta}(x)\Big)^2.$$

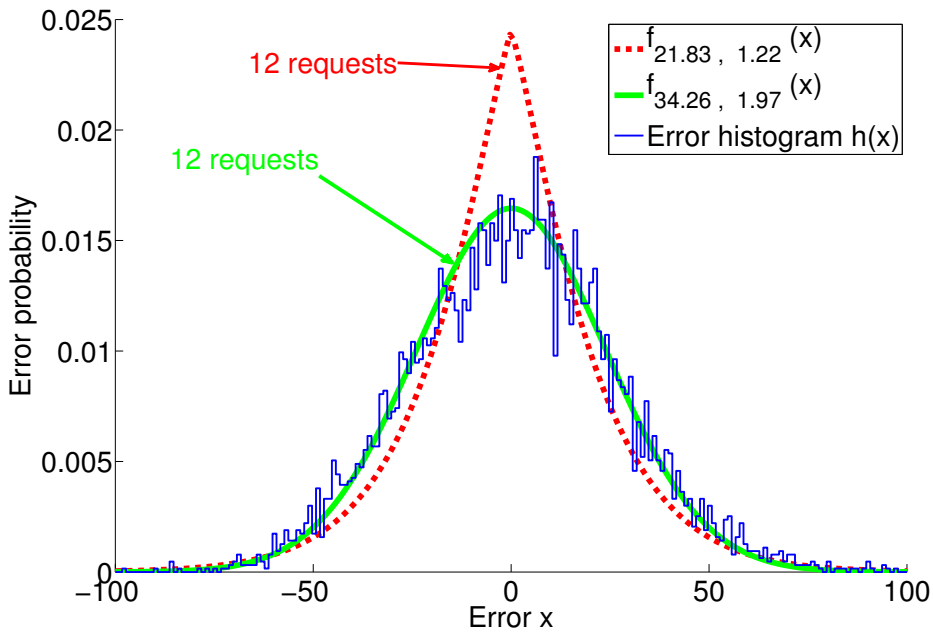


Figure 8.6: Two distributions modelling the true error histogram. They both allow a transmission with the minimum rate (corresponding to 12 turbodecoder requests) whereas one is far better fitted than the other.

The SSD, because of the square, penalizes high differences between the histogram and its model. Moreover this distance does not take into account the amplitude of the error x, *i.e.*,a difference between the model and the histogram would cost the same price for a low or high error x.

If we want to avoid the high difference penalization, one can replace the square by a power lower than 1 ($\frac{1}{2}$ for example):

$$d_{\frac{1}{2}SD}(h, f_{\alpha,\beta}) = \sum_x \left| h(x) - f^*_{\alpha,\beta}(x) \right|^{\frac{1}{2}}.$$

Another classical distance is the Kullbach-Leiber distance (KLD) [Kullback, Leibler, 1951], which is designed for pdf similarity description:

$$d_{KLD}(h, f_{\alpha,\beta}) = \sum_x h(x) \log \frac{h(x)}{f^*_{\alpha,\beta}(x)},$$

or

$$d_{KLD}(h, f_{\alpha,\beta}) = \sum_x \frac{1}{2} \left(h(x) \log \frac{h(x)}{f^*_{\alpha,\beta}(x)} + f_{\alpha,\beta}(x) \log \frac{f^*_{\alpha,\beta}(x)}{h(x)} \right),$$

for its symmetric version. Contrary to SSD, the KLD penalizes high ratios (and not high differences). In other words the KLD would advantage the distribution which performs a good fitting on the queue of distribution (where $h(x)$ is lower, *i.e.*,when x is higher). In

the following d_{KLD} denotes the symmetric KLD.

In order to test the reliability of these metrics, for all of the bands of several frames of different video sequences at various quantization steps, we propose to make the following experiment. Having an error histogram (offline) $h(x)$, and the Discover estimated Laplacian $f_{\hat{\alpha},1}$, we run the turbodecoding of a same side information, with a large number (600) of different GG $f_{\alpha,\beta}$ and measure the required rates for the current band. The different distributions are generateed randomly around the initial Laplacian pdf. Besides, for each of the distribution we measure the distance to the true histogram.

For each distance we count the number of times when the following assertion is true over the whole database:

$$\forall \alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathbb{R}, \qquad d(f_{\alpha_1,\beta_1}, h) \leq d(f_{\alpha_2,\beta_2}, h) \Leftrightarrow r_{\alpha_1,\beta_1} \leq r_{\alpha_2,\beta_2}, \tag{8.4}$$

where $r_{\alpha,\beta}$ is the required rate for turbodecoding the SI under the model $f_{\alpha,\beta}$, all the decoded frames having the same quality.

The obtained statistics indicate that the KLD is the most appropriate metric among the three proposed measures, but without obtaining a constant and acceptable percentage of Equation (8.4) truthfulness. Indeed, the validation of assertion in Equation (8.4) could reach 95% in some cases but 80% in other configurations (band, sequence, etc.)[2]. Therefore, it could be interesting to investigate more deeply the obtained results by displaying the 3D surfaces: $(x = \alpha, y = \beta, z = r_{\alpha,\beta})$ and $(x = \alpha, y = \beta, z = d_{KLD}(h, f_{\alpha,\beta}))$.

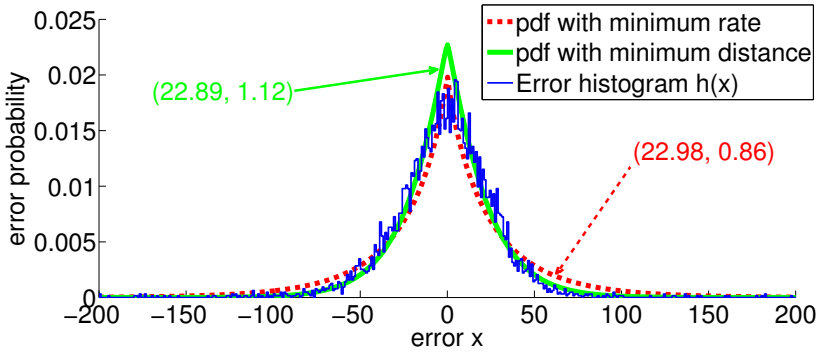
In Figures 8.7 and 8.8, we present two typical examples. Before commenting them, a little explanation of what is displayed is needed. Firstly, we generate a set of 600 random parameter couples (α, β) in a relatively wide but realistic range (based on many observations): $0 < \beta < 2$ and $20 < \alpha < 90$.

For each of the 600 couples (α, β) we measure the required rate (denoted by $r_{\alpha,\beta}$) by the turbodecoding of the corresponding band, with the *a priori* information calculated based on $f_{\alpha,\beta}$. Moreover, for each couple, we measure the distance $d_{KLD}(h, f_{\alpha,\beta})$. For both figures, we present the results as explained in the following:

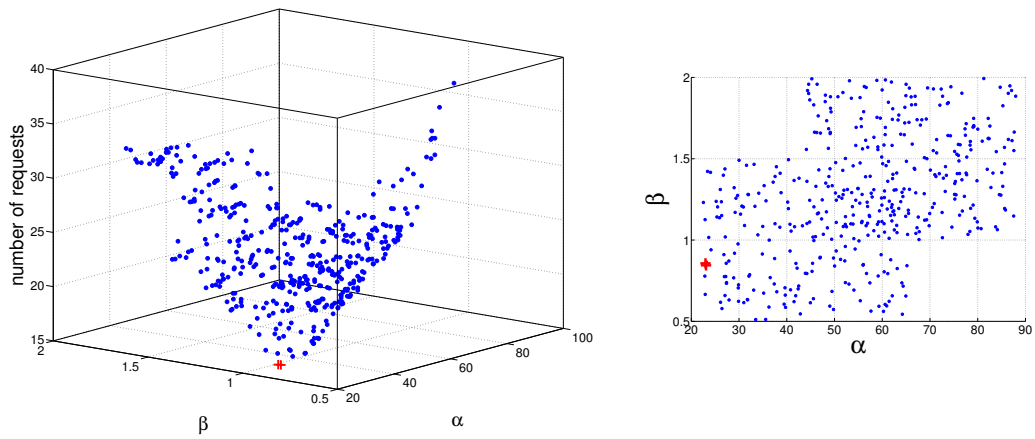
- (a): the representation of the histogram (blue), the pdf (or one of the pdfs) which achieves the lowest required rate (in red) and the pdf which reaches the minimum distance to the histogram (in green).
- (b): 3D representation of the obtained rate (expressed in number of requests) as a function of the coefficients α and β. On the left, the cloud of points is represented in 3D, on the right, an horizontal projection is illustrated. The crosses represented in red correspond to couples which reach the minimum rate, *i.e.*,the optimal distribution models.
- (c): similar 3D representation as in (a) with the KLD instead of the rate. The red crosses still correspond to the couples which obtain the minimum rates, and the green point indicates the couple which achieves the minimum distance, *i.e.*,the estimation of the best distribution model (which is not necessarily the real optimal model).

[2]Thus, precise average statistics would not give interesting information.

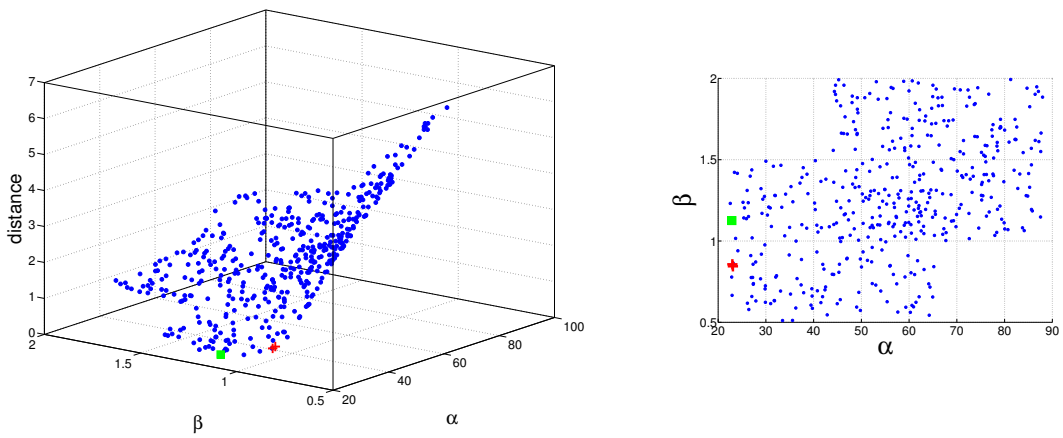
Figure 8.7: Example of experimental results obtained for *soccer* sequence. The pdf distribution which obtained the minimum rate (respectively the minimum KLD distance to the histogram) are in red (respectively in green).



(a) Estimation of the true error distribution $h(x)$

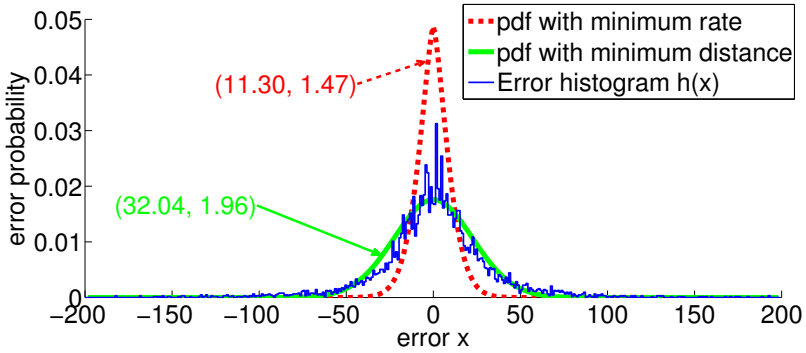


(b) rate in function of α and β (3D representation on the left, and up view on the right). Red crosses correspond to the minimum rates.

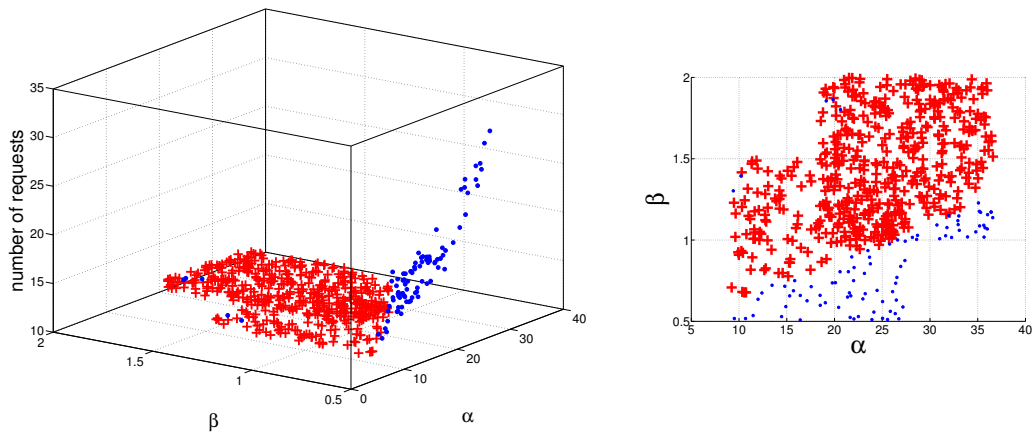


(c) KLD distance between the model and the error histogram as a function of α and β (3D representation on the left, and up view on the right). Red crosses correspond to the minimum rates and the green square is the minimum distance.

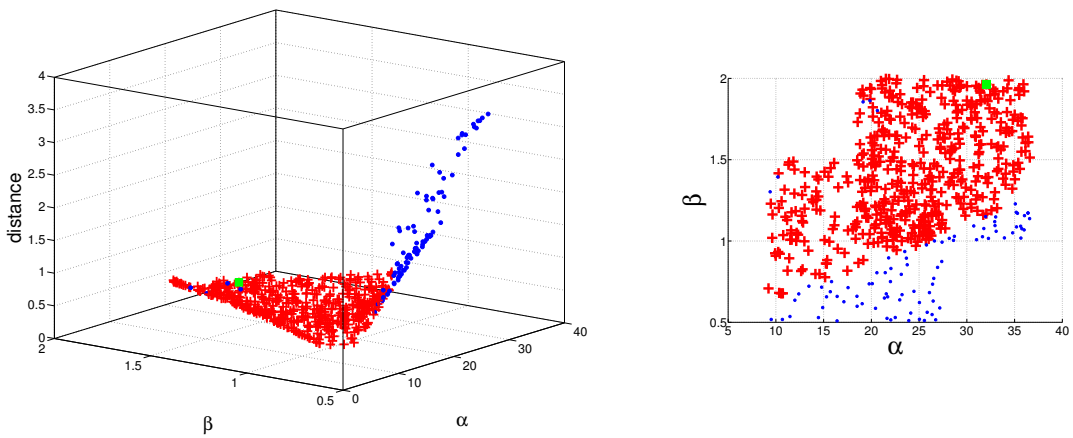
Figure 8.8: Example of experimental results obtained for *mobile* sequence. The pdf distribution which obtained the minimum rate (respectively the minimum KLD distance to the histogram) are in red (respectively in green).



(a) Estimation of the true error distribution $h(x)$



(b) rate in function of α and β (3D representation on the left, and up view on the right). Red crosses correspond to the minimum rates.



(c) KLD distance between the model and the error histogram as a function of α and β (3D representation on the left, and up view on the right). Red crosses correspond to the minimum rates and the green square is the minimum distance.

8.3.2.2 Discussion

If we analyse the two cases displayed in Figures 8.7 and 8.8, we observe than in the first one the green distribution (with the minimum KLD) does not achieve the minimum rate (*i.e.*,the green point is out of the red point zone in subplot (c)) while in the second the minimum distance pdf achieves a minimum rate.

The main observation we can make about these results is the following. In one case (Figure 8.7) only two distributions (very similar) achieve a minimum rate and a small modification of the optimal α or β implies a rate improvement. In other words, the (α, β) determination strongly impacts on the turbodecoding rates. It can be seen by observing the red and green pdf which are quite similar, and lead to totally different rates. Figure 8.8 shows a totally different case of figure: the red zone (corresponding to the minimum rate) is very wide which means that almost all tested couples (exactly 85%) achieve the minimum number of requests. It can also be observed in Figure 8.8 (a), where the plotted pdf are very different, but achieve a similar rate.
The second observation happens in any band of every sequence, it is not a isolated example. This could explain the limits of GG refinement that we described at the beginning of Section 8.3.

8.3.3 Conclusion

The conclusion of these experiments is firstly that the GG model works always better or at least similarly to a Laplacian one, which justifies the proposition of using a GG model. Moreover, it was observed that sometimes a better fitted distribution improves the performance. However, it is also observed that refining the model is not necessarily the only criterion that matters for improving the RD performances, the choice of the distance being probably also to be further studied. Moreover, these observations may also be explained by the fact that the correlation is not stationary over the frame, and a memoryless model cannot be the best solution. This was tackled in some very recent works by using Hidden Markov Model (HMM) [Toto-Zarasoa *et al.*, 2010] or particle filtering [Stankovic *et al.*, 2010]. In addition to spatially correlated models, informed models (*e.g.* , using hash) may probably better respond to this problem.

Chapter 9

Side information quality estimation

The study presented in Part II has shown that distributed video coding performances strongly depend on the quality of the side information. Indeed, an estimation Y (performed at the decoder) close to the original frame X would require a few parity bits for error correction. The purpose of side information construction is to build the best estimation. The problem studied in this chapter is the meaning of "best estimation". The most popular distortion measure for side information quality is the PSNR with respect to the reference WZ frame, but nothing assures that this represent the best evaluation in this specific framework, and in this chapter we try to show why. In Section 9.1 we present some tests which point out the limits of a PSNR measure. In Section 9.2, we describe the existing measures for SI quality estimation in a DVC context, and in Section 9.3 we present some novel measures. Then, we compare state-of-the-art measures and the proposed ones (in Section 9.5) in several experimental conditions.

Contents

9.1 Motivations **208**
9.2 State-of-the-art **208**
9.2.1 PSNR metric 208
9.2.2 SIQ 209
9.3 Proposed metric **210**
9.3.1 Generalization of the SIQ 210
9.3.2 A Hamming distance based metric 211
9.4 Methodology of metric comparison **212**
9.5 Experimental results **214**
9.5.1 Common side information features 214
9.5.2 The reasons why the PSNR is commonly used 214
9.5.3 The limits of the PSNR 217
9.6 Conclusion **223**

9.1 Motivations

PSNR metric is used almost always when dealing with side information estimation. The literature shows that, often, a PSNR gain for the side information results a PSNR gain (or rate saving) for the decoded video. However it is known that this is not always the case. For example, Kubasov, in [Kubasov, 2008], presented one case where one side information has a better PSNR than another, but after decoding, the second one has a better reconstruction for a lower rate. In other words, there exist some cases where the PSNR metric is not reliable for predicting the impact on the end-to-end rate-distortion performances.

In this chapter, we extend the Kubasov study and propose a more complete analysis of PSNR metric performance. Moreover we test the Kubasov metric, SIQ, and our metric based on the Hamming distance.

In Kubasov thesis manuscript [Kubasov, 2008] the two side informations were generated by a motion interpolation method and a simple spatial interpolation method. Here, we present another "artificial" example. The video sequence is *foreman*, in CIF format, at 30 frames per second. For the frame number 10, we generate two side informations. One is constructed with the Discover interpolation of frame 9 and 11 (Figure 9.1 (a) and first line of Table 9.1). The PSNR of this estimation is 29.05 dB. The second side information (Figure 9.1 (b) and second line of Table 9.1) was built by adding a uniform random noise on the original frame in order to obtain the same PSNR (29.04 dB). Then both side informations were turbodecoded with the same conditions (QI=8 for the WZ quantization). Results are presented in Table 9.1 and show that in spite of an equivalent PSNR, the two side informations do not obtain the same decoding performances. Indeed, the Discover interpolation allows to obtain a decoded frame at a PSNR of 39.29 dB, using a rate of 137.28 kb, while the artificial noisy estimation needs more rate (192.46 kb) and leads to a poorer decoded image (35.40 dB).

Table 9.1: An example of the limits of PSNR metric as a side information quality measure.

Type of SI	PSNR of the SI (dB)	rate (kb)	decoded PSNR (dB)
DISCOVER interpolation	29.05	137.28	39.29
Original + Artificial noise	29.04	192.46	35.40

In this particular case, we can see that the PSNR does not give a good information on the evaluation of the SI quality. The purpose of this chapter is to determine if this example is isolated and rarely happens in practice, or on the contrary if we can better understand when the PSNR can be trusted and when it presents its limits (and in this case, if the proposed metrics are reliable).

9.2 State-of-the-art

9.2.1 PSNR metric

The Peak-Signal-Noise-Ratio (PSNR) was developed to estimate the image quality in general, in presence of a reference. For example, it is used to estimate the noise in an image, I, while comparing it to its original, I_{ref}. In case of classical images (*i.e.*,the pixel values

(a) (b)

Figure 9.1: The two side informations of the example in Table 9.1: (a) DISCOVER29.05 dB and (b) artificial noise 29.04 dB.

have a dynamic of 255) its expression reads:

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{MSE}\right)$$

where MSE is the Mean Square Error between the image and its reference:

$$\text{MSE} = \frac{1}{N_{\text{width}} \times N_{\text{height}}} \sum_{\mathbf{p}\in[\![1,N_{\text{height}}]\!]\times[\![1,N_{\text{width}}]\!]} \Big(I(\mathbf{p}) - I_{ref}(\mathbf{p})\Big)^2.$$

The PSNR is known to be a first order estimator of human visual perception, because of the mean-square error. Indeed, human vision is more sensible to high magnitude differences, and MSE penalizes high errors (in opposition for example with MAD, the Mean Absolute Difference). This metric is then commonly adopted to evaluate image and video quality, even though it is far from being perfect and its drawbacks have been largely discussed in the literature [Wang, Bovik, 2009][Girod, 1993]. For example, an image shifted to one pixel left would have a very poor PSNR, although human vision would see no difference. Furthermore, in video coding, PSNR does not take any temporal aspects into account, despite the fact that our perception is very sensible to motion activity.
Whereas PSNR presents some limits to estimate the decoded video quality, it is not the point of our study, and then we keep the PSNR to measure the distortion at the output of the decoder. Here we study the limits of the PSNR in its role of estimating the side information quality. We investigate why PSNR would be justified whereas there is no visual consideration before turbodecoding.

9.2.2 SIQ

In his PhD thesis [Kubasov, 2008], Kubasov proposes a novel metric that he called *Side Information Quality (SIQ)*. Instead of using a squared error, he defines the SIQ metric using a squared root:

$$\text{SIQ} = 10\log_{10}\left(\frac{255^2}{\frac{1}{N_{\text{width}}\times N_{\text{height}}}\sum_{\mathbf{p}\in[\![1,N_{\text{height}}]\!]\times[\![1,N_{\text{width}}]\!]}\left|I(\mathbf{p}) - I_{ref}(\mathbf{p})\right|^{\frac{1}{2}}}\right). \tag{9.1}$$

The choice of using the root comes from the following argument. At the channel decoder, the side information is used to produce the log-likelihood ratio (LLR):

$$LLR = \log \frac{p(x=0)}{p(x=1)}.$$

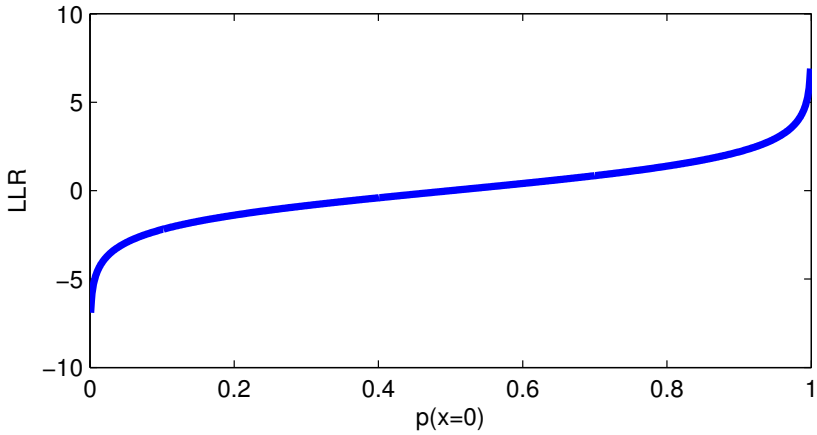


Figure 9.2: LLR as a function of $p(x=0)$.

The plot in Figure 9.2 displays the aspect of the LLR in function of the probability $p(x=0)$. One can remark that the LLR is almost constant and near to zero for a wide probability range (between 0.1 and 0.9). The consequence of it, is that for high and medium errors, the decoder obtains almost the same LLR value, what is the opposite of MSE behaviour. After this observation, the use of power $\frac{1}{2}$ becomes justified, since the main property of $x \to x^{\frac{1}{2}}$ function is that it is almost constant for high x and vary a lot for low values of x.

Kubasov in his manuscript has tested the SIQ and showed for one example that the SIQ could be more reliable than PSNR. In addition to the fact that SIQ was not deeply tested and proved to be reliable, Kubasov does not investigate why the PSNR fails sometimes and why it keeps being a reliable measure other times; two problems that we propose to tackle in this chapter.

9.3 Proposed metric

9.3.1 Generalization of the SIQ

The SIQ idea of changing square in the PSNR formula immediatly leads us to propose metrics based on other power than $\frac{1}{2}$. In fact, it would be interesting to test any kind of metrics, SIQ_a, given by a

$$\mathrm{SIQ}_a = 10 \log_{10} \left(\frac{255^2}{\frac{1}{N_{\mathrm{width}} \times N_{\mathrm{height}}} \sum_{\mathbf{p} \in [\![1, N_{\mathrm{height}}]\!] \times [\![1, N_{\mathrm{width}}]\!]} \left| I(\mathbf{p}) - I_{ref}(\mathbf{p}) \right|^a} \right)$$

with $a \in (0, 1]$ and with the same notations than in Section 9.2.2. While it is obviously impossible to test all the values for a, we propose to retain two specific values:

- $a = 1$, which correspond to the l_1-norm commonly used in signal processing. We call the associated metric SIQ_1.

- $a = \frac{1}{3}$, in this case we try to further enhance the difference between small error values. The metric associated to $\frac{1}{3}$ is called $\text{SIQ}_{\frac{1}{3}}$.

For uniformization reasons, the original SIQ metric is denoted by $\text{SIQ}_{\frac{1}{2}}$ in the following. After this direct Kubasov's work generalization, in the next section, we propose to develop metrics which are more adapted to the turbodecoding procedure.

9.3.2 A Hamming distance based metric

In our DVC framework, after transform and quantization, the WZ frame is transposed into bitplanes. Each bitplane is encoded successively (the most significant coming first). For each bitplane, the decoder receives a first sample of parity bits and starts the decoding algorithm of the corresponding bitplane of the side information. If the error probability is too high ($> 10^{-3}$), the decoder requests one more set of parity bits, restarts the decoding and so forth.

The PSNR and the SIQ sum the difference in the spatial domain with more or less importance given to high errors. However a difference in the spatial domain is far from what the channel decoder is sensible to. Indeed, a Slepian-Wolf decoder requires parity information as long as the error probability of the bitstream remains too high. Between the comparison in the spatial domain, and the sensibility of the turbodecoder, there are two important blocks: a transformation and a quantization.

With this new measure, we propose to take into account the structure of the WZ coder. Thus, we propose a metric based on a Hamming distance between the side information bitstream and the original bitstream. If $\overline{I}$ and $\overline{I}_{ref}$ are the transformed and quantized versions (at $\text{QI} = qi$) of respectively the SI and the reference image, b denotes the band, bp the bitplane, c the coefficient and N_{bits} the total number of binary numbers in the frame decomposition, the proposed *Hamming Side Information Quality (HSIQ)* metric is given by:

$$\text{HSIQ}(qi) = 10 \log_{10} \left(\frac{1}{\frac{1}{N_{\text{bits}}} \sum_b \sum_{bp} \sum_c \overline{I}(b, bp, c) \oplus \overline{I}_{ref}(b, bp, c)} \right) \tag{9.2}$$

where $\oplus$ denotes the binary addition operator.

The advantage of this metric is that it is very close to the turbodecoder behaviour. The difference with the PSNR and SIQ, is that the HSIQ measures the required rate and not the distortion, which is exactly what the turbodecoder does when establishing an error probability threshold at 10^{-3}.
Moreover, another advantage of the HSIQ metric is that it depends on the quantization of the WZ frame, which can be very interesting for estimating SI quality for specific quantization conditions.

9.4 Methodology of metric comparison

In this section, we introduce the methodology used for estimating the reliability of the existing and the proposed metrics. Contrary to decoded video quality metrics which have to be compared with human subjective experiments for their reliability tests, side information quality measures must be correlated with the rate-distortion performance of the codec. The rate-distortion performance is measured with a couple $(R, d) \in \mathbb{R}^+ \times \mathbb{R}^+$, which is not obvious to compare with another rate-distortion couple in the 2D space. Figure 9.3 illustrates the fact that, having only two points does not give an order information. Indeed, both possibilities shown inFigures 9.3 (b) and (c) are conceivable. In the following, we introduce a theoretical environment allowing to compare two couples under a rate distortion model.

The ordering between RD curves has more chances to succeed if we have several rate-distortion points. For example, in Figure 9.4 (a) and Figure 9.4 (b), one can determine the better curve. On the contrary, in Figure 9.4 (c), it is not obvious to see which curve is better than the other. That is why we use the commonly adopted Bjontegard metric.
In [Bjontegaard, 2001] Bjontegaard proposed a method for comparing two rate-distortion curves. This technique needs 4 points for each curve and calculates the area between them and can deliver two types of comparison: the Bjontegaard PSNR gain yields the average gain in PSNR (dB) for the same number of bits, while the Bjontegaard bit savings yields the average savings in bits for the same resulting PSNR.

In the following, the Bjontegaard comparison function is denoted by $bjm\,(.,.)$ whose inputs are two sets of 4 rate-distortion couples (the first input is the reference). Since the Bjontegaard comparison result can be given in both rate diminution or PSNR gain in dB, we choose arbitrarily, for the following, to compare the different curves in terms of rate saving percentage (of the second input with respect to the first input). In other words a rate-distortion curve $(R_i^1, d_i^1)_{i=1\ldots4}$ is under another $(R_i^2, d_i^2)_{i=1\ldots4}$ if the Bjontegaard metric $bjm\left((R_i^1, d_i^1)_{i=1\ldots4}, (R_i^2, d_i^2)_{i=1\ldots4}\right) \leq 0$.

Based on the Bjontegaard comparison, we can now define an equivalence relation between two sets of 4 RD points, referred to as "RD sets", (and their associated schemes) $\forall\ (R_i, d_i)_{i=1\ldots4} \in (\mathbb{R}^+ \times \mathbb{R}^+)^4$,

$$(R_i^1, d_i^1)_{i=1\ldots4} = (R_i^2, d_i^2)_{i=1\ldots4} \Leftrightarrow bjm\left((R_i^1, d_i^1)_{i=1\ldots4}, (R_i^2, d_i^2)_{i=1\ldots4}\right) = 0 \tag{9.3}$$

and similarly we define an order relation between RD sets through:

$$(R_i^1, d_i^1)_{i=1\ldots4} \leq (R_i^2, d_i^2)_{i=1\ldots4} \Leftrightarrow bjm\left((R_i^1, d_i^1)_{i=1\ldots4}, (R_i^2, d_i^2)_{i=1\ldots4}\right) \leq 0. \tag{9.4}$$

The reflexivity, transitivity and symmetry can be easily proven. Having this equivalence relation, the corresponding class of equivalence can be defined as:

$$\forall\ (R_i, d_i)_{i=1\ldots4} \in \left(\mathbb{R}^+ \times \mathbb{R}^+\right)^4, \qquad [(R_i, d_i)_{i=1\ldots4}] = \left\{(R_i^{eq}, d_i^{eq})_{i=1\ldots4} \in \left(\mathbb{R}^+ \times \mathbb{R}^+\right)^4 \text{ such as } bjm\left((R_i, d_i)_{i=1\ldots4}, (R_i^{eq}, d_i^{eq})_{i=1\ldots4}\right) = 0\right\}. \tag{9.5}$$

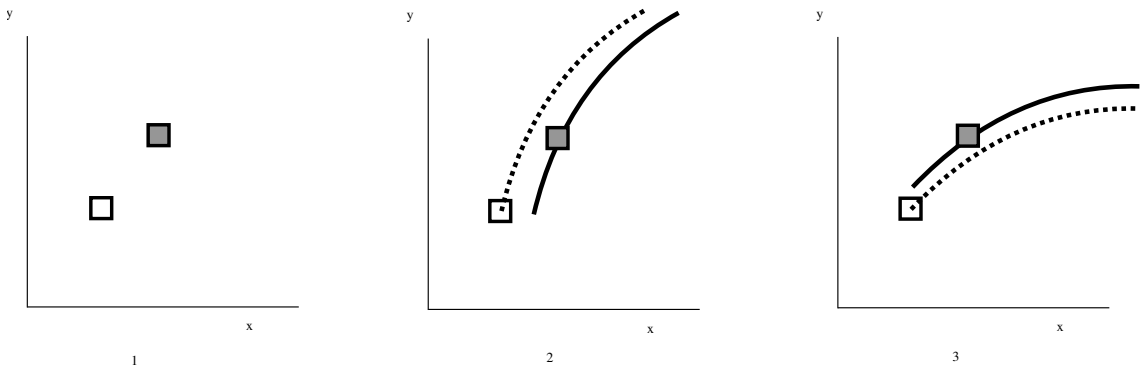


Figure 9.3: It is difficult to compare two rate-distortion points in the 2D space without any additional information.

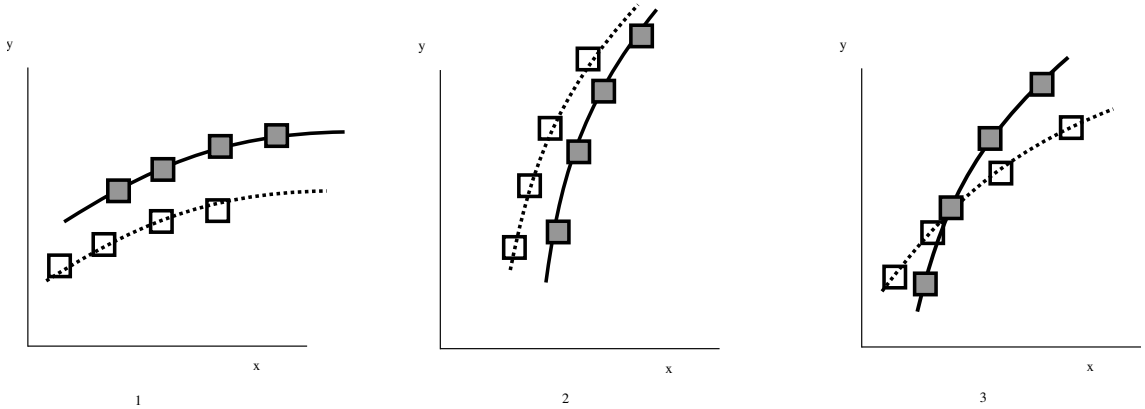


Figure 9.4: Having 4 RD points it is possible in the most part of the cases to determine the order between RD points of the same curves, excepting when the curves are crossed. In this case we propose to use the Bjontegaard metric to come to a decision.

The set of the equivalence classes is denoted by $\mathcal{RD}$. One can now introduce an order relation between the equivalent classes of $\mathcal{RD}$:

$$\begin{aligned}&\forall[(R_i^1, d_i^1)_{i=1\ldots4}] \in \mathcal{RD}, \quad \forall[(R_i^2, d_i^2)_{i=1\ldots4}] \in \mathcal{RD},\\ &[(R_i^1, d_i^1)_{i=1\ldots4}] \leq [(R_i^2, d_i^2)_{i=1\ldots4}] \Leftrightarrow bjm\left((R_i^1, d_i^1)_{i=1\ldots4}, (R_i^2, d_i^2)_{i=1\ldots4}\right) \leq 0. \end{aligned} \tag{9.6}$$

Having now a possibility to compare two RD sets, we want to link this order relation to the side information quality estimation issue. If n and m are two non-zero integers, we denote by $\mathcal{I}_{n,m}$ the set of images of height n and width m. Let us define a decoder function *dec* which has two images as input.

The first image I_0 is the original frame which is encoded and decoded based on the second input image (I_1) as side information. The *dec* function associates these two images to one set of 4 rate distortion couples ($[(R_i^1, d_i^1)_{i=1\ldots4}]$) obtained by encoding the original frame at 4 quantization steps and decoding it with I_1 as side information. More precisely the rate-distortion couple gives the rate R required to obtain the decoded frame with a distortion d using the side information, I_1.

Thanks to this theoretical setting, we are now able to state whether a side information, I_1, is better or not than another, I_2. We only need to compare $dec(I_0, I_1) = [(R_i^1, d_i^1)_{i=1...4}]$ and $dec(I_0, I_2) = [(R_i^2, d_i^2)_{i=1...4}]$ with the order relation defined above. Because final turbodecoding performances optimization constitutes our principal goal, this quality order between two estimations is the *real* order, *i.e.*,the order we want to model, with the quality metrics (as PSNR, SIQ or HSIQ) which are functions from $(\mathcal{I}_{n,m})^2$ to $\mathbb{R}$ ($\mathcal{M}$ is the set of the quality metrics).

To measure the reliability of the metrics, we introduce the following *confidence criterion.* A metric $m \in \mathcal{M}$ respects the confidence criterion if:

$$\forall (I_0, I_1, I_2) \in \mathcal{I}_{n,m}, \quad dec(I_0, I_1) \leq dec(I_0, I_2) \Leftrightarrow m(I_1, I_0) \leq m(I_2, I_0) \tag{9.7}$$

In the following, we test the different metrics with respect to this confidence criterion over different test sequences in our database, for different experimental conditions.

9.5 Experimental results

9.5.1 Common side information features

Experiments presented in this Section 9.5 consist in testing the reliability of the different metrics described previously. This reliability is given by statistics computed on different experimental databases, which are composed by side informations of different qualities. Then the "confidence criterion" is calculated by counting the number of times the metric estimates the good quality order. The results are given by a final percentage which indicates a confidence measure of the metrics (more details will be given in Section 9.5.2 and 9.5.3). To be relevent, tests must be run on representative sets of side informations. First, for one WZ frame, its estimations must be numerous (100 in our tests), then the quality range between the best and the worst SI must be relatively wide (almost $2 - 3$ dB in PSNR). The estimation generation methods are explained in Sections 9.5.2.1 and 9.5.3.1. Finally, the database must contain real errors. This is why, in this section we analyse the different origins of the errors in a side information.

In DVC, the methods to generate side information for the Wyner-Ziv frames are numerous, but the ones which are commonly used are motion estimation based algorithms. Based on the already decoded frames, these methods use motion information to build the estimation of the WZ frame. Even if the approaches differ (interpolation, extrapolation, fusion see Part II for more details) the general structure is based on a reference frame compensation. Then, the two types of errors under consideration in that type of side information are the quantization of the reference frames, and the motion estimation errors (essentially block artefacts).

9.5.2 The reasons why the PSNR is commonly used

9.5.2.1 Experiment settings

The first experiments correspond to the case where the estimation is generated with reference frames compressed at the same level of quantization. This is the case in a scheme such as Discover[DISCOVER-website, 2005]. The SI database is generated for each of

the first 100 frames of *breakdancer*, *book arrival*, and *outdoor* sequences[1], and for each quantization step (reference frames are quantized at QP 31, 34, 37 and 40). Let I_0 be one original WZ frame of a test sequence. Let $\widetilde{I_1}$ and $\widetilde{I_2}$ be two quantized reference frames (at a fixed QP). To generate the database, we first estimate the backward and forward motion vector fields respectively between $\widetilde{I_1}$ and I_0, and between $\widetilde{I_2}$ and I_0. They are denoted by $\mathbf{u}_1$ and $\mathbf{u}_2$. Assuming, as explained in the previous section, that the estimation error comes from inaccuracies in some vectors, we generate the N different side informations, $\widehat{I_k^{si}}$, of the experimental database, by introducing iid errors on a random number of motion vectors. At the end, the PSNR of the obtained SI is controlled in such a way that the PSNR range of the databased is not wider than Δ which is a threshold fixed in advance. The procedure is detailed in *Algorithm 1.*
Once all the database is created, for each frame of each sequence at each quantization level, all the SIs are turbodecoded at 4 QI (4,5,6,7)[2]. In other words, we compare $\forall k \in \{1, \ldots, N\}$,

$$(R_i^k, d_i^k)_{i=1\ldots4} = dec(\widehat{I_0, I_k^{si}}).$$

Then, we are able to compute the statistics measuring the reliability of the metrics. For each metric $m \in \mathcal{M}$, we compute the percentage cases when the following equivalence is satisfied $\forall k \in \{1, \ldots, N\}$ and $\forall l \in \{1, \ldots, N\}$:

$$m(\widehat{I_k^{si}}) \leq m(\widehat{I_l^{si}}) \Leftrightarrow bjm\left((R_i^k, d_i^k)_{i=1\ldots4}, (R_i^l, d_i^l)_{i=1\ldots4}\right) \leq 0 \tag{9.8}$$

The results are presented and discussed in the next section.

9.5.2.2 Discussion

The tests were run for 3 sequences: *breakdancer*, *outdoor*, *book arrival* (512×384 resolution) at four quantizations steps for the key frames (31, 34, 37, 40). Each of the generated side information is decoded at four QI (4, 5, 6, 7) in order to obtain the class $[(R_i, d_i)_{i=1\ldots4}]$. The database contains 100 different side informations, with a value of Δ (which determines the maximum difference in PSNR between the estimation of the database) equal to 3 dB (PSNR). For the generation of a specific estimation, the maximum number of affected blocks is equal to 100 and the maximum error applied to a vector field is 10 pixels.

The results are presented in Table 9.2. The percentages correspond to the number of times when the Equation (9.8) is verified. One can remark than the PSNR, the SIQ and the HSIQ obtain similar results. The three metrics seem to be reliable for this type of database.
In other words, since the reference frames are similarly quantized and the estimation error comes from motion vector imprecision, the different side information qualities are well estimated by the PSNR (and by the several SIQ_a metrics and by the HSIQ). Therefore, these experiments do not cast doubt on the majority of the papers using PSNR to measure their improvement on a reference method, because they are in this case of figure (similar reference frames and motion estimation/compensation interpolation methods).
However, the limits of PSNR exist, as in the examples presented in Section 9.1, and we

[1]These three sequences have been chosen because they present very different characteristics.
[2]The chosen QI are high beacause a too coarse WZ quantization would not be appropriate with the SI quality range, and would make their turbodecoding diverge.

Input: The original frame I_0, the two quantized reference frames $\widetilde{I}_1$ and $\widetilde{I}_2$.
Output: a set of N side informations, $\left(\widehat{I_i^{si}}\right)_{i=1:N}$, of different quality;

$N_{affectedBlocksMax}$ - maximum number of affected motion vectors;
E_{Max} - maximum error applied to motion vectors;
N_{Blocks} - number of blocks per vector fields;
Δ - maximum dB difference between the SI PSNR of the database;

begin
Initialization: calculation of the two motion vector fields with a motion estimation (me)

$$\mathbf{u}_1 = me\left(I_0, \widetilde{I}_1\right) \text{ and } \mathbf{u}_2 = me\left(I_0, \widetilde{I}_2\right)$$

$\widehat{I_0^{si}} = \frac{1}{2}\left(\widetilde{I}_1\left(\mathbf{u}_1\right) + \widetilde{I}_2\left(\mathbf{u}_2\right)\right)$; `/* initial SI */`

i=1;
while $i \leq N$ **do**
$\mathbf{u}_1^i \leftarrow \mathbf{u}_1$;
$\mathbf{u}_2^i \leftarrow \mathbf{u}_2$;
$N^i_{affectedBlocks} \leftarrow rand() * N_{affectedBlocksMax}$; `/*` $rand()$ `gives a random number between 0 and 1 (uniform) */`
for $j = 1 : N^i_{affectedBlocks}$ **do**
$n_j \leftarrow floor(rand() * N_{Blocks})$; `/* random block selection */`
$e_j \leftarrow 2 * (rand() - 0.5) * E_{Max}$; `/* random error */`
$e'_j \leftarrow 2 * (rand() - 0.5) * E_{Max}$; `/* random error */`
$\mathbf{u}_1^i(b_{n_j}) += (e_j, e'_j)$;
$\mathbf{u}_2^i(b_{n_j}) += -(e_j, e'_j)$;
end
$\widehat{I_i^{si}} = \frac{1}{2}\left(\widetilde{I}_1\left(\mathbf{u}_1^i\right) + \widetilde{I}_2\left(\mathbf{u}_2^i\right)\right)$; `/* Average of the 2 motion compensated frames */`
Validation: keep the generated SI if its $PSNR$ is in the acceptable range **if** $|PSNR(\widehat{I_i^{si}}) - PSNR(\widehat{I_0^{si}})| \leq \frac{\Delta}{2}$ **then**
save $\widehat{I_0^{si}}$;
$i++$;
end
end
end

Algorithm 1: Side information database generation with identically quantized reference frames

	breakdancer				*outdoor*				*book arrival*				
QP	31	34	37	40	31	34	37	40	31	34	37	40	Avg
PSNR	90.1	87.6	90.4	87.3	89.9	93.2	92.0	90.5	90.9	91.7	92.0	90.0	90.5
SIQ_1	89.9	89.2	89.3	87.0	89.1	92.5	93.1	89.0	92.2	91.7	92.0	91.1	90.5
$SIQ_{\frac{1}{2}}$	89.7	88.7	89.4	86.0	89.0	92.5	93.0	88.9	91.6	92.2	92.2	90.6	90.3
$SIQ_{\frac{1}{3}}$	89.0	86.0	87.5	87.0	88.3	92.2	92.7	88.7	90.0	91.1	92.5	89.8	89.6
HSIQ	89.1	86.6	87.7	86.5	88.9	93.6	93.3	89.0	90.4	91.0	92.9	90.1	90.0

Table 9.2: Percent of veracity of the confidence criterion of Equation (9.8) for several sequences and several quantization steps for the reference frames used to generate the side information databases.

shall see in the next section in which context they may happen, and if the other metrics manage to estimate correctly the side information quality.

9.5.3 The limits of the PSNR

The study of the previous section has shown that in a database where the quantization of the reference frames was the same for all the N estimations, the PSNR gives good reliability results (as good as the SIQ_a and the HSIQ). The previous database corresponds to the case where all of the N estimations have a similar type of error, block artifacts and similar quantization. Nevertheless, the counterexamples provided in Section 9.1 were obtained with side information presenting very different types of error. The DISCOVER interpolation has block artifacts (high and localized errors), the spatial error and the noisy frame have a small error affecting almost all the pixels. In this section, we aim at constructing a database with different types of errors. This database needs to be realistic, it should represent error configurations similar to those obtained with actual DVC interpolation schemes.

9.5.3.1 Experiment settings

In the next section, the side information generation method is similar to the one of Section 9.5.2, but differs in the fact that the reference frames are not quantized with the same QP. In other words, the QP is also a random variable. In order to keep a fixed Δ in PSNR between the maximum and the minimum values, the $N_{affectedBlocksMax}$ depends on the quantization of the reference frames. In other words, in the database, good quality key images would generate estimations strongly affected by the vectors errors, and on the contrary, coarse reference frames based estimations would be very slightly affected by the additional motion vector errors. The method for side information generation is given in *Algorithm 2*.
This database is realistic and not artificial. Indeed, schemes involving key frames quantized at different QP can be easily considered. For example, in case of multiview coding, the quantization can be different for each camera. Furthermore, it can also be the case in the sequences where the QP is changed during the coding process.

9.5.3.2 Discussion

As for Section 9.5.2, tests were run for three video sequences: *outdoor*, *book arrival* and *breakdancer* (512×384). All of the 100 generated side informations are turbodecoded at four QI (4, 5, 6 and 7) in order to determine for each of them the class $[(R_i, d_i)_{i=1...4}]$ they

Input: The original frame I_0, the two original reference frames I_1 and I_2.
Output: a set of N side informations, $\left(\widehat{I_i^{si}}\right)_{i=1:N}$, of different quality;

$N_{affectedBlocksMax}(QP)$ - maximum number of affected motion vectors which depends on the QP of the key frames;
E_{Max} - maximum error applied to motion vectors;
N_{Blocks} - number of blocks per vector fields;
Δ - maximum dB difference between the SI PSNR of the database;

begin
  i=1;
  while $i \leq N$ **do**
    Key frame quantization: QP $\leftarrow$ randomly 31, 34, 37 or 40
    Quantization of reference frames at QP $\rightarrow \widetilde{I_1}$, $\widetilde{I_2}$
    Initialization: calculation of the two motion vector fields with a motion estimation (me)

$$\mathbf{u}_1 = me\left(I_0, \widetilde{I_1}\right) \text{ and } \mathbf{u}_2 = me\left(I_0, \widetilde{I_2}\right)$$

    $\widehat{I_0^{si}} = \frac{1}{2}\left(\widetilde{I_1}\left(\mathbf{u}_1\right) + \widetilde{I_2}\left(\mathbf{u}_2\right)\right)$; `/* initial SI */`
    $\mathbf{u}_1^i \leftarrow \mathbf{u}_1$;
    $\mathbf{u}_2^i \leftarrow \mathbf{u}_2$;
    $N_{affectedBlocks}^i \leftarrow rand() * N_{affectedBlocksMax}(QP)$; `/*` $rand()$ `gives a random number between 0 and 1 (uniform) */`
    for $j = 1 : N_{affectedBlocks}^i$ **do**
      $n_j \leftarrow floor(rand() * N_{Blocks})$; `/* random block selection */`
      $e_j \leftarrow 2 * (rand() - 0.5) * E_{Max}$; `/* random error */`
      $e'_j \leftarrow 2 * (rand() - 0.5) * E_{Max}$; `/* random error */`
      $\mathbf{u}_1^i(b_{n_j}) += (e_j, e'_j)$;
      $\mathbf{u}_2^i(b_{n_j}) += -(e_j, e'_j)$;
    end
    $\widehat{I_i^{si}} = \frac{1}{2}\left(\widetilde{I_1}\left(\mathbf{u}_1^i\right) + \widetilde{I_2}\left(\mathbf{u}_2^i\right)\right)$; `/* Average of the 2 motion compensated frames */`
    Validation: keep the generated SI if its $PSNR$ is in the acceptable range **if** $|PSNR(\widehat{I_i^{si}}) - PSNR(\widehat{I_0^{si}})| \leq \frac{\Delta}{2}$ **then**
      save $\widehat{I_0^{si}}$;
      $i++$;
    end
  end
end

Algorithm 2: Side information database generation with different quantized reference frames

Sequence	*breakdancer*	*outdoor*	*book arrival*	Average
PSNR	66.09	61.65	74.99	67.57
SIQ_1	92.27	91.90	95.66	93.27
$SIQ_{\frac{1}{2}}$	90.83	91.88	95.27	92.66
$SIQ_{\frac{1}{3}}$	90.53	91.81	94.79	92.37
HSIQ	90.84	93.82	93.68	92.78

Table 9.3: Percent of veracity of the confidence criterion of Equation (9.8) for several sequences and for the second database with different types of errors.

belong to.
Once the database is obtained (side informations generated and their decoded quality calculated), first experiments lead us to obtain statistics presented in Table 9.3. They are the percentages of veracity of the confidence criterion (Equation (9.8)) over the different sets of side informations. While SIQ_a and HSIQ seem to keep being reliable, one can easily observe that this is no longer the case for the PSNR metric. The PSNR gives the right quality order between two side informations in only 2 cases out of 3 (HSIQ and SIQ are right in more than 90% of the cases). These results highlight that in some cases, PSNR is far from being completely reliable.

In the following, we investigate one particular case[3] and try to analyse why the PSNR is sometimes wrong in SI quality estimation.
Then, let us focus on the side information database of frame 3 of *outdoor* sequence. First we sort the 100 side informations in the decoding performances growing order and we number them in this order. In other words, i and j are two natural numbers between 2 and 100, we have

$$dec(I_0, \widehat{I_i^{si}}) \leq dec(I_0, \widehat{I_j^{si}}) \Leftrightarrow \qquad i \leq j$$

with the order relation defined by Equation (9.6). For each side information, $\widehat{I_j^{si}}$, we calculate its *relative rate saving (RRS)*, which is the rate decrease percentage (in the sense of Bjontegaard metric) comparing to $\widehat{I_{j-1}^{si}}$ added with the RRS of $\widehat{I_{j-1}^{si}}$:

$$\begin{aligned} &RRS(\widehat{I_1^{si}}) = 0 \\ &\forall i > 1, \qquad RRS(\widehat{I_i^{si}}) = bjm\left((R_i^i, d_i^i)_{i=1\ldots4}, (R_i^{i-1}, d_i^{i-1})_{i=1\ldots4}\right) + RRS(\widehat{I_{i-1}^{si}}). \end{aligned} \tag{9.9}$$

In other words, the RRS value for the ith side information corresponds to the cumulated rate saving with respect to the side information of lowest quality ($\widehat{I_1^{si}}$). In Figure 9.5 (a), we plot the RRS values. Between the lowest and the best side information we observe a RRS difference of approximately 4% which allows to say that the database is significantly wide from the point of view of the decoding quality.
Figures 9.5 (b)-(f) present the plots of respectively the PSNR, SIQ_1, $SIQ_{\frac{1}{2}}$, $SIQ_{\frac{1}{3}}$ and HSIQ for the same order. In other words, a quick look on these figures permits to see if the metric has a general growing appearance and then preserves the real quality order

[3] All of what is presented in the following is not an isolated exception and similar behaviours are observed all over the different databases. Thus, the following can be seen as a general trend and is very revealing of what exactly happens while measuring the side information quality with different techniques.

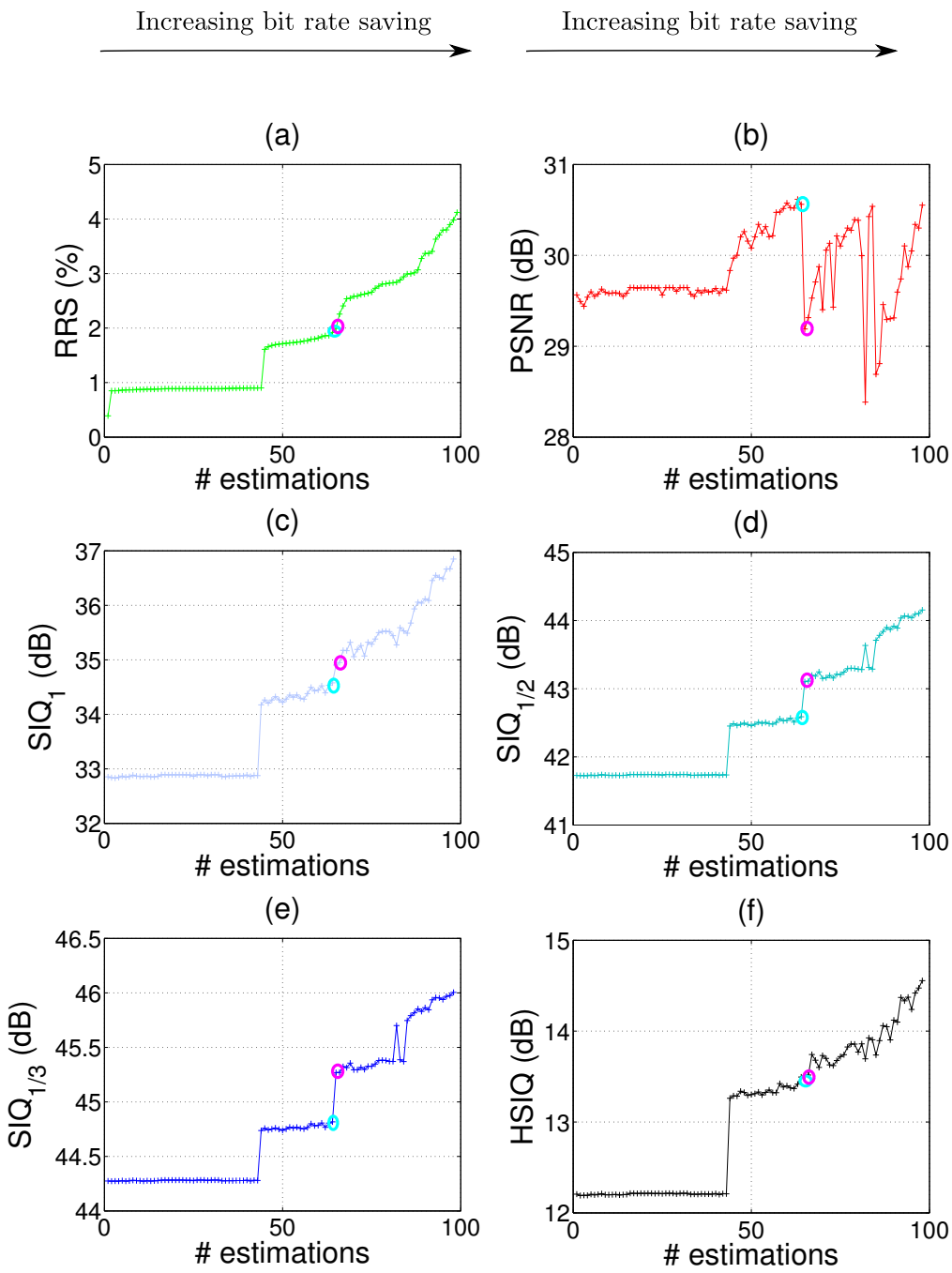


Figure 9.5: Metric values as a function of the number of estimations for frame 3 of *outdoor* sequence (512×384). The estimations are sorted in the decoded performance growing order (real quality). Cyan and purple circles indicate the examples illustrated in Figures 9.7, 9.8, 9.9, 9.10, 9.11 and 9.12.

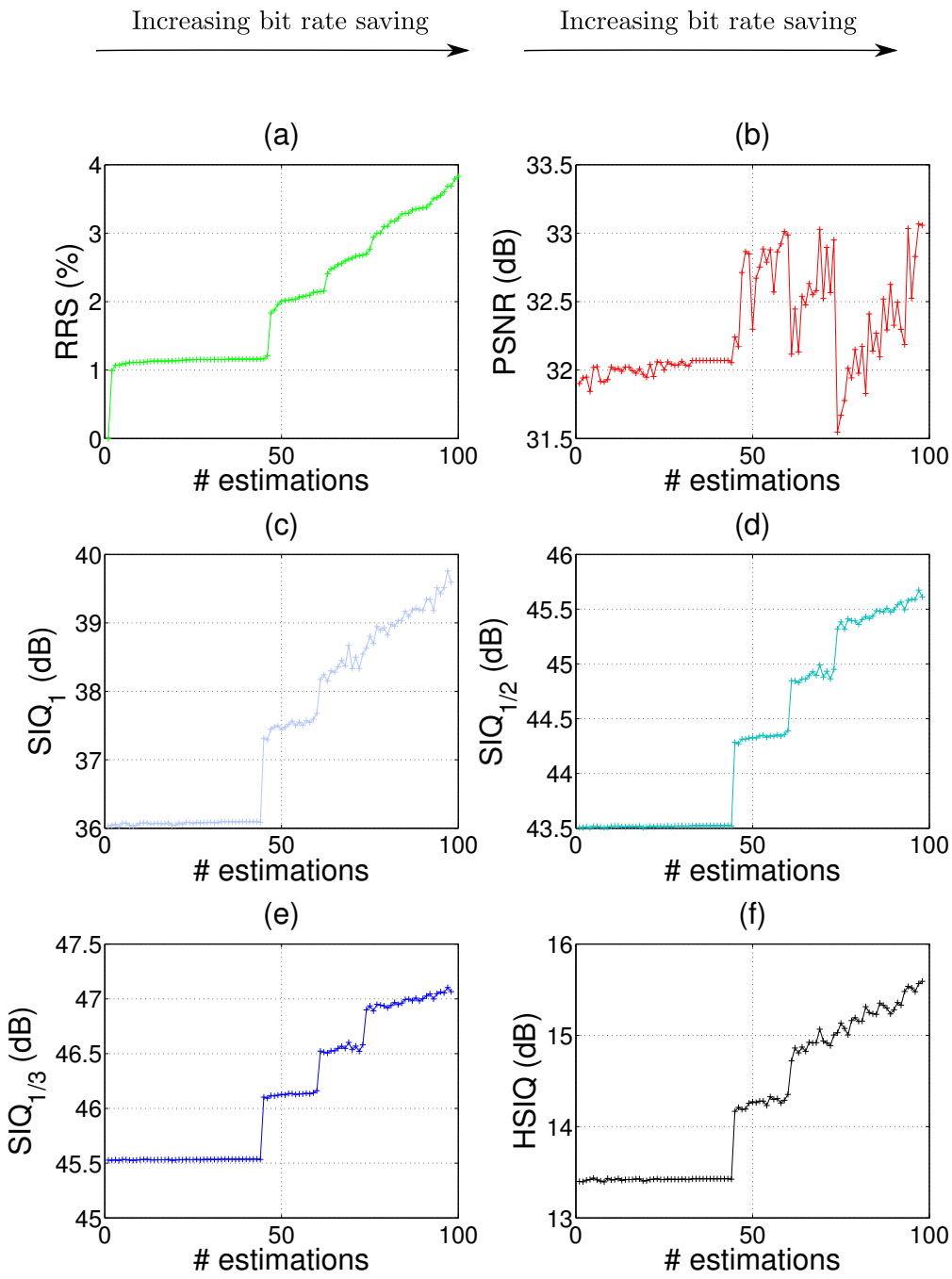


Figure 9.6: Metric values as a function of the number of estimations for frame 3 of *book arrival* sequence (512 × 384). The estimations are sorted in the decoded performance growing order (real quality).

of Figure 9.5 (a). It is thus easy to see that HSIQ and SIQ_a have an acceptable growing behaviour, similar to the RRS evolution, but on the other hand, PSNR does not preserve the ordering relationship since two consecutive estimations can have a negative variation of more than 1 dB instead of an improvement. Moreover, one can notice that SIQ_1 (Figure 9.5 (c)) and HSIQ (Figure 9.5 (f)) are the two metrics which evoluate the most similarly to the RRS behaviour (Figure 9.5 (a)), especially during the second part of the plot. One more time, this phenomenon happens for each of the tested databases, as the reader can see in another example (Figure 9.6, *book arrival*, frame 3).

Let us focus once again on a particular example which is revealing of what often happens. In the following we study the case of $\widehat{I_{64}^{si}}$ and $\widehat{I_{65}^{si}}$ (resp. cyan and purple circles in Figure 9.5). This study is motivated by the fact that, even though $RRS(\widehat{I_{64}^{si}}) < RRS(\widehat{I_{65}^{si}})$, the PSNR of the SI predicts the opposite order, *i.e.*,$PSNR(\widehat{I_{64}^{si}}) > PSNR(\widehat{I_{65}^{si}})$, and with a very high gap of more than 1.4 dB. SIQ_a and HSIQ predict the right order for these estimations, and then, a developed study of this example may be interesting as it can lead us to better understand the limits of PSNR.
Firstly, we propose to look at the side information images themselves. In Figures 9.7 (a) and (b), one can see the two estimations. One can easily remark that the block artifact errors are more numerous in $\widehat{I_{65}^{si}}$ than in $\widehat{I_{64}^{si}}$. Indeed, the random number of affected block, N_{Blocks}, is 30 for the estimation 64 and is 198 for the estimation 65. Moreover, $\widehat{I_{64}^{si}}$ has been constructed with reference frames quantized at a QP of 37 while $\widehat{I_{65}^{si}}$ is based on key frames compressed with a QP of 34. In other words, the two estimations both present distortion coming from key frame quantization and motion errors, but without the same proportions. Therefore, let us analyse the error image associated to the different metrics, in order to understand how the two types of error are taken into account by the measure. Since the PSNR is calculated with a SSD, we show in Figures 9.8 (a) and (b) the square error image. One can see that the square error only brings out high errors as blocking artifacts, and quantization error is thus not visible. This explains why the PSNR of $\widehat{I_{65}^{si}}$ is so much larger than the PSNR of $\widehat{I_{64}^{si}}$.
On the contrary, if we look at Figures 9.9 (a) and (b), which display the absolute error, one can perceive the quantization error in estimation error of the 64th SI. It is more obvious in Figures 9.10 (a) and (b) for the absolute error with a power of $\frac{1}{2}$, where the quantization error is almost as highly taken into account as the block errors. One can also remark that the quantization error is higher in the left image (QP 37) which explains that $SIQ(\widehat{I_{64}^{si}}) < SIQ(\widehat{I_{65}^{si}})$. Finally, this observation is even more visible for a power of $\frac{1}{3}$ (Figures 9.11 (a) and (b))
Then for a $qi = 1$ we plot the Hamming difference images band by band and bitplane by bitplane for the two estimations (Figures 9.12 (a) (b)). These images show that the number of different bits (white points) is visually similar in both estimation decompositions, which means that HSIQ also takes into account both types of errors with the same weight.

Visual results are interesting because they give an explanation of what happens when the PSNR fails. We can remark that the PSNR metric "counts" the number of high errors present in the image, mainly coming from blocking artifacts. On the other hand, the different SIQ_a metrics count the errors by more or less taking into account the error

intensity. Indeed, the lower the a value is, the less the magnitude of the error is considered. As a consequence, for the lowest values (as $\frac{1}{3}$ can be), the measure almost only "counts" the number of pixel where the estimation and the original differs, which is not exactly what corresponds to the decoder behaviour. It can be seen by looking at the appearance of the curves in Figures 9.5 and 9.6. Though the $\text{SIQ}_{\frac{1}{3}}$ is growing (see (d)), its evolution does not fit the RRS evolution of (a), and is more like a succession of 4 levels which corresponds to the 4 quantization steps of the reference frames. In other words, the $\text{SIQ}_{\frac{1}{3}}$ too much takes into account quantization errors. This remarks may also be done for the $\text{SIQ}_{\frac{1}{2}}$ which present a similar 4 levels looking despite its growing evolution. Under this consideration, SIQ_1 curves seem to have the more corresponding looking among the different SIQ_a.

Having also a quite acceptable looking, HSIQ make a compromise, because it considers the magnitude error information only when an error in a bitplane propagates to the next bitplane. In other words, when the numbers of block errors are of the same order of magnitude for two side informations, the PSNR can give a good estimation of the quality, but if the error types are different, *i.e.*,a highly concentrated error or a diluted error, the PSNR would disadvantage the highly concentrated error whereas it would be more easily corrected by the turbodecoder.

9.6 Conclusion

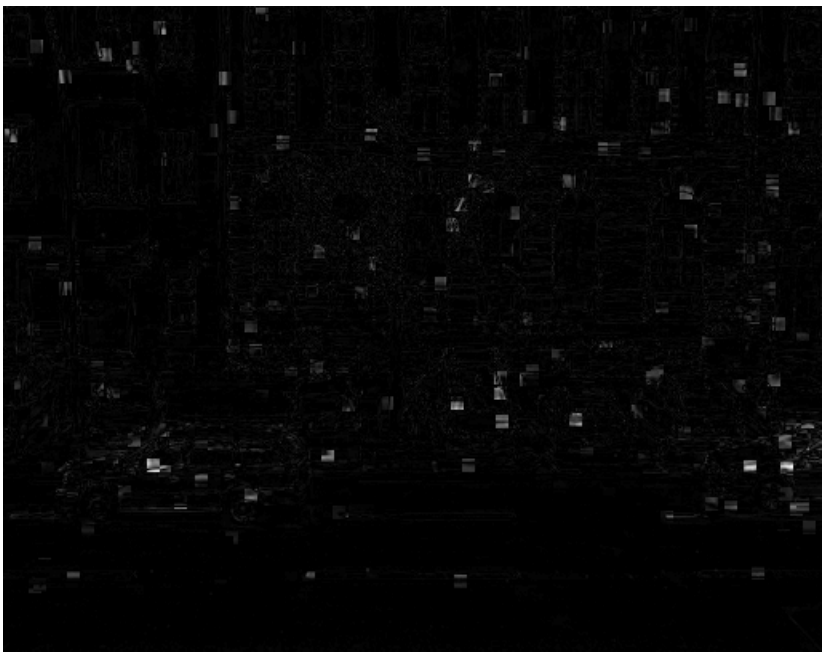
In this chapter we firstly demonstrate that PSNR metric, despite of an acceptable behaviour in some usual situations (like monoview DISCOVER configuration), presents important limits for comparing two side information qualities, especially when these one provide different types of errors (concentrated or diluted).
On the contrary, the family of SIQ_a metrics and the proposed hamming based HSIQ measure, obtained acceptable statistics and proved than it may be interesting to use measures which better correspond the turbodecoding behavior (the LLR for the SIQ_a, and the transform+quantization structure with the HSIQ).
In the results drawn in this chapter, the SIQ_a and HSIQ obtained similar performances. Both of these metrics seem to be adapted for measuring side information quality in a DVC context. An even more elaborated study should be necessary to differentiate which is the most reliable metric. Some remarks can however be made. Indeed, whereas the SIQ_a has proven to have good statistics over the tested database, the choice of the a coefficient has a strong importance. It was not verified in the statistics results of Table 9.3, but it is visible in the appearance of the SIQ_a curves in Figures 9.5 and 9.6. As it was discussed in the previous sections SIQ_1 seems to be the most adapted metric among the SIQ_a, as adapted as the HSIQ, which obtains acceptable statistics and also fits the RRS evolution. Moreover, HSIQ depends on the quantization and may be appropriate for a finer quality estimation, depending on the decoding conditions.

(a) $\widehat{I_{64}^{si}}$, 1.86% (RRS)

(b) $\widehat{I_{65}^{si}}$, 1.92% (RRS)

Figure 9.7: Zoom on the estimations 64 and 65 (respectively cyan and purple circles in Figure 9.5) and their corresponding RRS measure.

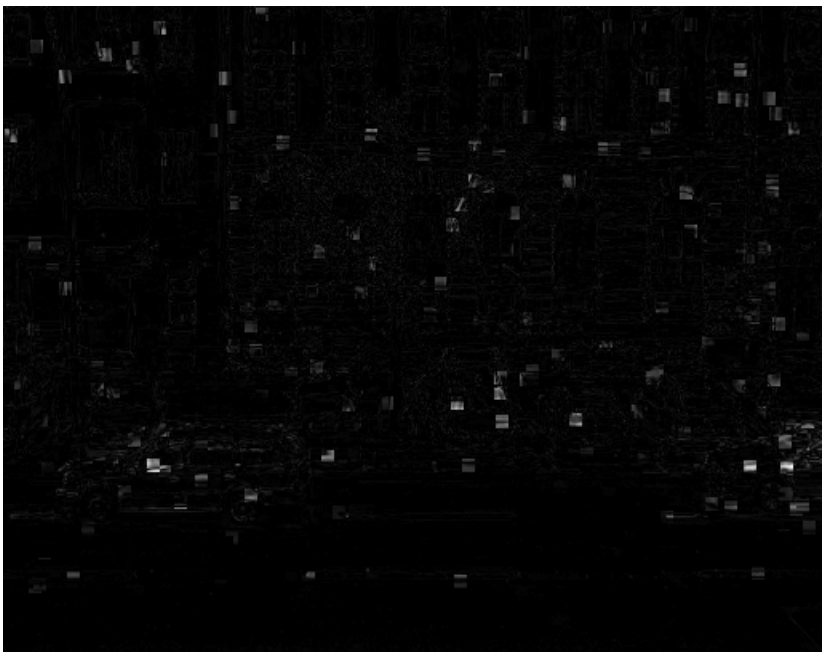
(a) $(\widehat{I_{64}^{si}} - I)^2$, 30.63 dB (PSNR)



(b) $(\widehat{I_{65}^{si}} - I)^2$, 29.20 dB (PSNR)

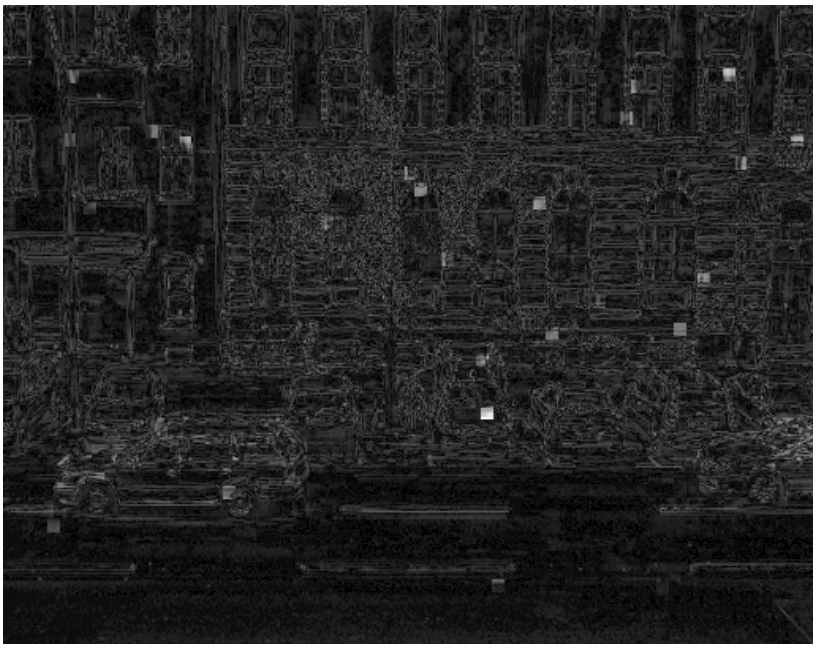
Figure 9.8: Zoom on the pixel domain error image associated to the PSNR measure, for the estimations 64 and 65 (respectively cyan and purple circles in Figure 9.5).

(a) $|\widehat{I_{64}^{si}} - I|$, 34.55 dB ($\mathrm{SIQ}_1$)

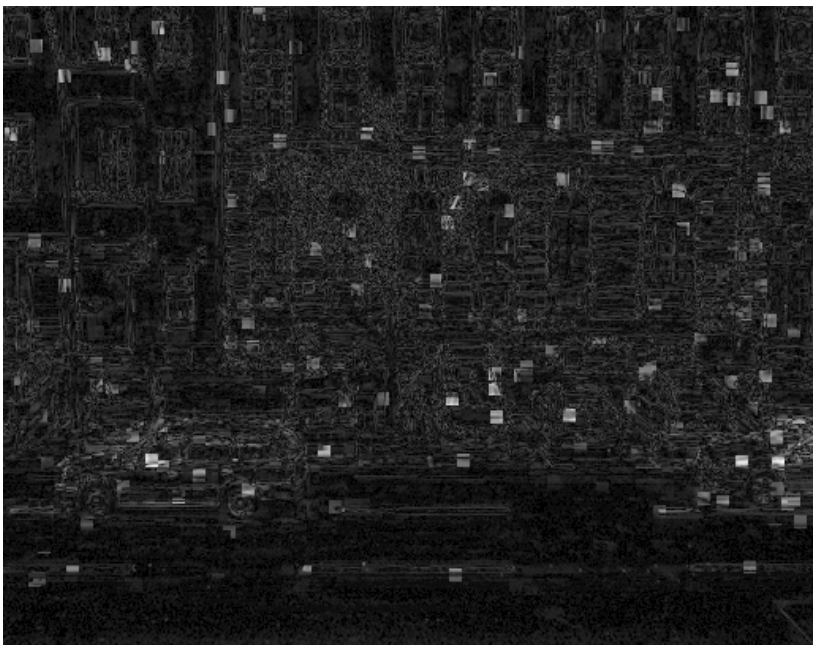


(b) $|\widehat{I_{65}^{si}} - I|$, 34.90 dB ($\mathrm{SIQ}_1$)

Figure 9.9: Zoom on the pixel domain error image associated to the SIQ_1 measure, for the estimations 64 and 65 (respectively cyan and purple circles in Figure 9.5).

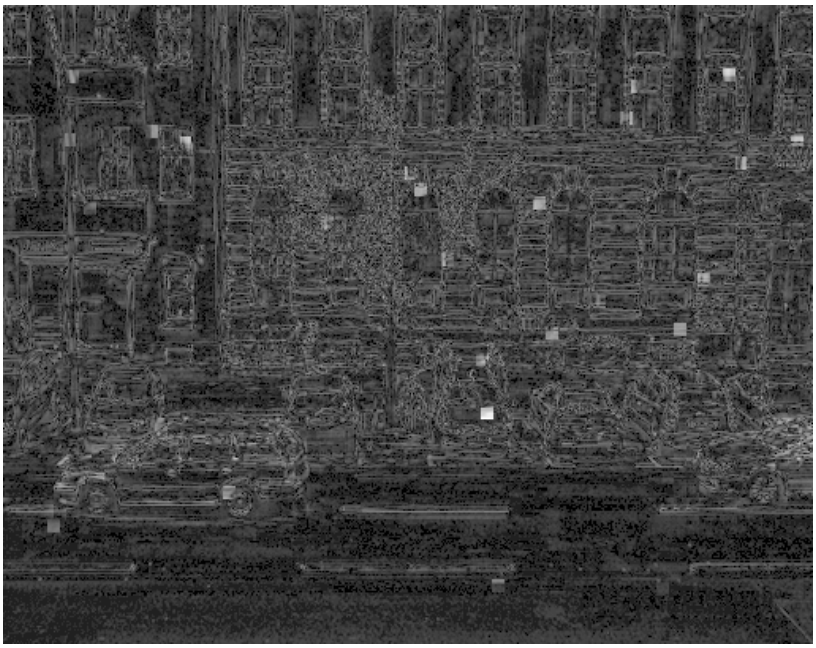


(a) $|\widehat{I_{64}^{si}} - I|^{\frac{1}{2}}$, 42.61 dB ($\mathrm{SIQ}_{\frac{1}{2}}$)

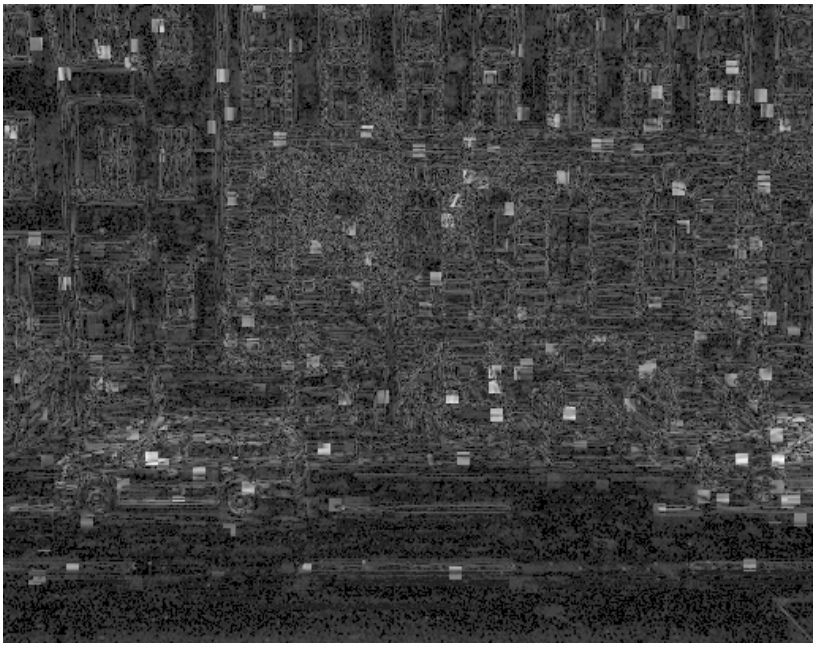


(b) $|\widehat{I_{65}^{si}} - I|^{\frac{1}{2}}$, 43.12 dB ($\mathrm{SIQ}_{\frac{1}{2}}$)

Figure 9.10: Zoom on the pixel domain error image associated to the $\mathrm{SIQ}_{\frac{1}{2}}$ measure, for the estimations 64 and 65 (respectively cyan and purple circles in Figure 9.5).

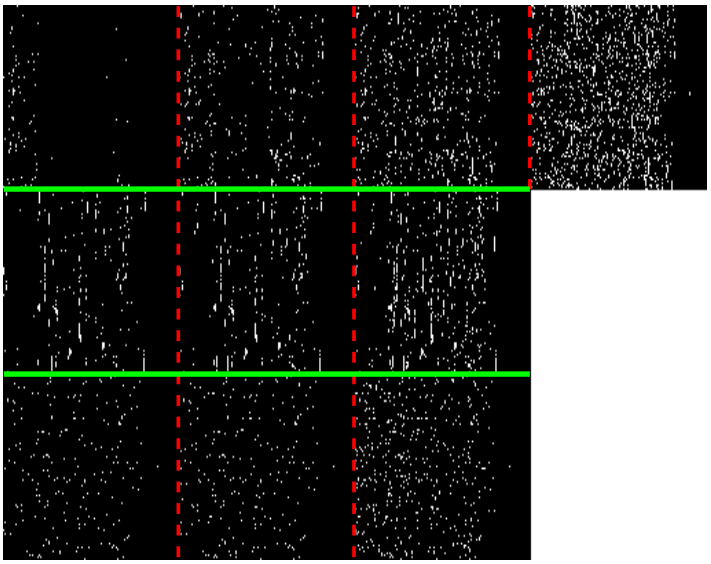


(a) $|\widehat{I_{64}^{si}} - I|^{\frac{1}{3}}$, 44.83 dB ($\mathrm{SIQ}_{\frac{1}{3}}$)

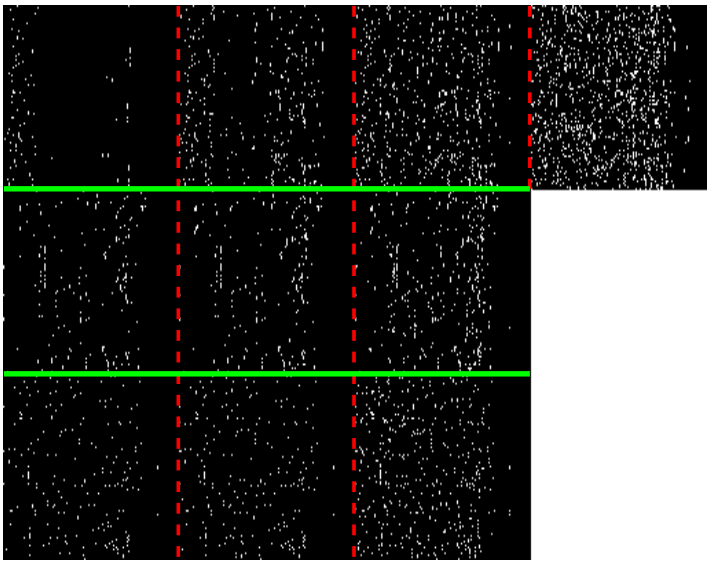


(b) $|\widehat{I_{65}^{si}} - I|^{\frac{1}{3}}$, 45.25 dB ($\mathrm{SIQ}_{\frac{1}{3}}$)

Figure 9.11: Zoom on the pixel domain error image associated to the $\mathrm{SIQ}_{\frac{1}{2}}$ measure, for the estimations 64 and 65 (respectively cyan and purple circles in Figure 9.5).



(a) $\overline{I_{64}^{si}} \oplus \overline{I}$, 13.49 dB (HSIQ)



(b) $\overline{I_{65}^{si}} \oplus \overline{I}$, 13.51 dB (HSIQ)

Figure 9.12: Zoom on the transform domain Hamming error image ($qi = 1$) for the estimations 64 and 65 (respectively cyan and purple circles in Figure 9.5) and the corresponding HSIQ value. Dashed red lines separate the bitplanes, while green plain lines separate the bands (the first band is in the first line, and the first bitplane is in first column).

Conclusion and future work

We firstly present here a summary of the contributions described in this PhD manuscript and we detail the future work resulting from them. Finally, we try to use the different works that we performed in order to build a more global vision of distributed video coding paradigm.

Summary of the thesis contributions

The main purpose of our thesis work was to tackle different issues brought by distributed video coding, and particularly in multiview settings. This has been concretised by the development of techniques which aimed at improving some modules of the Stanford scheme. Almost all of the proposed solutions have been developed and tested from a coder of DISCOVER type, that we have extended to multiview. Only the tests presented in Chapter 5 have been run with the ESSOR wavelet-based scheme.

A distortion model and its various applications to the coding scheme behavior analysis: firstly, we have proposed an expression for modelling the distortion of the WZ estimation error (Chapter 2). This model presents the main advantage of simplicity. Indeed, it separates the error coming from the motion or disparity estimation and the error due to the key frame quantization. In the tests, we have seen that the underlying approximations can generate a significant gap between the theoretical value and the true distortion. However, this gap is relatively constant and limited, and the model remains nontheless acceptable and useful for the target applications.
In Chapter 3, we have presented three applications of that proposed model. Firstly, we have studied the frame classification at the coder input, and we have proposed a new frame type repartition in the time-view space, which is less complex to encode and which outperforms the existing solutions. This new scheme was designed using the proposed model for determining the optimal WZ frame decoding order. Moreover, we have been able to analyze the error propagation phenomenon in case of entire frame losses. The model has predicted the coder behavior in such a case, and it was validated by the experiments. The model allowed us to work on a rate control algorithm at the encoder in order to definitely get rid of the return loop. If the proposed method leads to consequent losses of performances, these ones are of the same order of magnitude as the ones generated by the existing methods in the literature.

New approaches for side information generation: based on a detailed state-of-the-art review established in Chapter 4, we have proposed several techniques for side information quality enhancement at the decoder. The first of them (Chapter 5) has been

developed in collaboration with the members of the French ANR project ESSOR. Whereas this one still describes the motion by block, it is more precise since it manages the overlapping and empty regions. This chapter also shows in detail the proposed codec structure, developed within the project, and presents some rate-distortion results.
In Chapter 6, we present several interpolation and fusion methods which adopt a pixel approach. These interpolations are based on the DISCOVER interpolation structure and add two refinement modules, performed by using the Cafforio-Rocca and Miled algorithms (that we have adapted to the situation). Once the temporal and inter-view estimations generated, the proposed fusion methods merge the candidates by making a linear combination between the pixels, instead of a binary choice classically performed in the literature.
Based on the idea that some regions in the WZ image could not be estimated by the key frames at the decoder (rapid motion, occlusions, etc.) some schemes, called "hash-based" schemes, proposed to send to the decoder these regions hardly estimable or some information which helps the decoder to recover them. We have proposed in Chapter 7 a new approach for such schemes, by developping original techniques for hash information selection, and hash-based side information generation.

Zoom on the decoder: the study of the relation between the side information and the turbodecoding has appeared to us to be an interesting research issue, in the sense that it is one fundamental point of the distributed coding approach. This study led us to investigate two different problems. The first concerns the correlation noise estimation at the decoder, in order to calculate the *a priori* probabilities. The first observation was to remark that in the literature, a model refinement necessarily led to rate distortion improvements. Therefore, we have proposed a new type of model (Generalized Gaussian) instead of the classically adopted Laplacian one. Whereas the new model enhanced the turbodecoding efficiency for some sequences, as we predicted, there were some cases when a refinement did not lead to an improvement. We have then performed more advanced tests, and we have indeed verified that in some cases, almost all the tested models obtained the same performances since they remained at a frame level precision (or more exactly, a frequency band frame level).
The second problem highlighted by our work is the side information quality estimation. In almost all the works, even if the gains are validated by rate-distortion performances, the WZ estimation quality is estimated by the PSNR. In Chapter 9, we have shown (by extending the work initiated by Kubasov) that the PSNR failed in some situations. We have then proposed other metrics which remain more reliable for all the situations, because they are closer to the turbodecoder behavior. We however precise that this study on quality evaluation metrics does not put in question the results obtained in Part 2, where the estimations were compared using the PSNR measure, because they were performed in the situations where the PSNR is reliable.

Perspectives and future work

Based on the results of our contributions and based on the conclusions we have drawn from them, we detail here the different ideas which would be, to our opinion, interesting to investigate.

New extrapolation based multiview schemes containing less key frames: the

symmetric scheme proposed in Chapter 3 obtaining better results than the literature, it would be interesting to investigate more classification types involving even less key frames, and the side information techniques adapted to them. If the use of interpolation limits the extension of the distance between the reference frames (because interpolation is not competitive when the key frames are too far), we should think of performing extrapolations which do not become less efficient when this distance grows. This would necessitate to elaborate extrapolation methods for inter-view estimations, not existing nowadays. On the other hand, a frame loss could be dramatic for the performances. It would be thus interesting to study this phenomenon using an extension to multiview of our proposed rate-distortion model.

A rate control algorithm extended to multiview and less dependent of the offline parameters: once the rate-distortion model is extended to multiview, it would become possible to extend the proposed rate control algorithm to multiview. However, for both monoview and multiview configurations, it is necessary to work on a practical version of this algorithm. Indeed, the existing one is based on parameters which need to be estimated offline and which depend on the sequence. These parameters must be estimated online, directly by the encoder.

A better online adaptation for the dense interpolation methods: the results obtained in Chapter 6 led us to the following conclusions: the proposed methods can be very efficient in some situations, but do not improve the block-based DISCOVER approach in other cases. We think that this is due to a too high dependence to parameters, and that it would be interesting to build an online estimation solution for them.

Fusion methods based on shape recognition: after the exploration of linear fusions, it would certainly be beneficial to base the linear combination coefficient calculation on "object" considerations. In other words, it would be interesting to detect the objects in the scene, and thus predict the regions corresponding to high motion and occlusions.

Extension of the Generalized Gaussian model to the non spatially stationnary case: we have seen that in some situations, the performances keep unchanged for almost all the chosen parameters fixed for the Generalized Gaussian modelling the correlation noise. In other words, the distribution to model is not well chosen, and should be considered as non spatially stationary. Indeed, in an image, the correlation between side information and original image is not the same in all the regions, and it should be interesting to consider this phenomenon with a Generalized Gaussian distribution (or with the addition of several distributions, as it was performed in DSC framework [Bassi *et al.*, 2008] with Gaussian-Bernouilli-Gaussian models).

Applications for the proposed side information quality metrics: The study performed in this manuscript about the side information quality metrics do not go further than theoretical (but interesting) considerations. It would be thus beneficial to find some applications for these ideas, in order to improve the rate-distortion performances. For example, we could develop some side information generation methods in which the mean-square-error could be replaced by one of the proposed reliable metrics.

An ESSOR codec optimization in order to test our contributions with two different types of coders: even if we have presented some rate-distortion performances of the distributed video coding scheme ESSOR, we have seen that these performances were not yet optimized. For this objective, we should work on each of the codec modules and optimize it (WZ frame quantization, noise correlation, etc.). Once the coder available, we should then test the different contributions of this thesis with the ESSOR scheme. It should be interesting to observe the behavior of the proposed quality metrics with an LDPC decoder, or to test the Generalized Gaussian model performance on this same LDPC decoder, in the wavelet domain.

What future for distributed video coding?

Distributed video coding is a quite unusual research paradigm. Indeed, its novelty, its potential and the beauty of its underlying theoretical results contribute to make it very popular and to the fact that many research groups work for its coding performances improvement, which have the consequence that, in spite of the domain youth, the state-of-the-art is already weighty. However, this effervescence is nowadays being smoothed out. We see in some articles review that some researchers start to be skeptical about the distributed video coding potential. One one hand, the current results are not up to the ones expected, on the other hand, the complexity decreasing argument is less and less convincing. Indeed, the main justification of distributed video coding was initially to reduce the need of power at the encoders for some low-power systems (like cellphones), yet it is obvious that, with the efficiency improvements of the current processors, cellphones are rapidly going to be able to perform more and more heavy calculations.

However, we should not be pessimistic about the future of distributed video coding. Indeed, if the complexity argument is not convincing any more, there will always be one considerable advantage brought by a distributed approach: the suppression of the need of communication between cameras. It is very plausible that the technology progress will not rapidly sweep away this argument. Another reason for being optimistic about the future of distributed video coding is the enormous potential that it represents. For each module of its architecture, it is obvious that there remain many important improvements to do. For example, the side information generation techniques can still be enhanced, especially in the inter-view direction. An important issue of distributed video coding is the correlation noise estimation which needs to find the existing several stationarities. At least, if some researchers highlighted the limits of the Stanford scheme, it is nonetheless conceivable to invent another coding scheme, allowing to be closer to the theoretical conditions.

List of publications

Journal article

1. T. Maugey and B. Pesquet-Popescu, "Side information estimation and new symmetric schemes for multi-view distributed video coding," *J. on Visual Communication and Image Representation*, vol. 19, no. 8, pp. 589–599, Dec. 2008, special issue: Resource-Aware Adaptive Video Streaming.

Conference papers

1. T. Maugey, C. Yaacoub, J. Farah, M. Cagnazzo, and B. Pesquet-Popescu, "Side information enhancement using an adaptive hash-based genetic algorithm in a wyner-ziv context," in *Int. Workshop on Multimedia Sig. Proc. (MMSP)*, Saint-Malo, France, Oct. 2010.

2. M. Trocan, T. Maugey, E. Trames, F. JE, and B. Pesquet-Popescu, "Cs-reconstruction of multiview images using bootstrap-like disparity compensation," in *Int. Workshop on Multimedia Sig. Proc. (MMSP)*, Saint-Malo, France, Oct. 2010.

3. G. Petrazzuoli, T. Maugey, M. Cagnazzo, and B. Pesquet-Popescu, "Side information refinement for long duration gops in dvc," in *Int. Workshop on Multimedia Sig. Proc. (MMSP)*, Saint-Malo, France, Oct. 2010.

4. M. Trocan, T. Maugey, E. Tramel, J. Fowler, and B. Pesquet-Popescu, "Compressed sensing of multiview images using disparity compensation," in *Proc. Int. Conf. on Image Processing (ICIP)*, Sep Hong-Kong, 2010.

5. M. Trocan, T. Maugey, J. Fowler, and B. Pesquet-Popescu, "Disparity-compensated compressed-sensing reconstruction for multiview images," in *Int. Conf. on Multimedia and Expo. (ICME)*, Singapore, Aug 2010.

6. T. Maugey, J. Gauthier, B. Pesquet-Popescu, and C. Guillemot, "Using an exponential power model for wyner-ziv video coding," in *Proc. Int. Conf. on Acoust., Speech and Sig. Proc. (ICASSP)*, Dallas, Texas, USA, Mar 2010.

7. M. Cagnazzo, W. Miled, T. Maugey, and B. Pesquet-Popescu, "Image interpolation with edge-preserving differential motion refinement," in *Proc. Int. Conf. on Image Processing (ICIP)*, Cairo, Egypt, Nov. 2009.

8. T. Maugey, W. Miled, M. Cagnazzo, and B. Pesquet-Popescu, "Méthodes denses d'interpolation de mouvement pour le codage vidéo distribué monovue et multivue," in *Proc. GRETSI*, Dijon, France, Sep. 2009.

9. J. Gauthier, T. Maugey, B. Pesquet-Popescu, and C. Guillemot, "Amélioration du modèle statistique de bruit pour le codage vidéo distribué," in *Proc. GRETSI*, Dijon, France, Sep. 2009.

10. T. Maugey, W. Miled, M. Cagnazzo, and B. Pesquet-Popescu, "Fusion schemes for multiview distributed video coding," in *Proc. Eur. Sig. and Image Proc. Conference (EUSIPCO)*, Glasgow, Scotland, Aug. 2009.

11. W. Miled, T. Maugey, M. Cagnazzo, and B. Pesquet-Popescu, "Image interpolation with dense disparity estimation in multiview distributed video coding," in *Int. Conf. on Distributed Smart Cameras*, Como, Italy, Sep. 2009.

12. M. Cagnazzo, T. Maugey, and B. Pesquet-Popescu, "A differential motion estimation method for image interpolation in distributed video coding," in *Proc. Int. Conf. on Acoust., Speech and Sig. Proc. (ICASSP)*, Taipei, Taiwan, Apr. 2009.

13. T. Maugey, W. Miled, and B. Pesquet-Popescu, "Dense disparity estimation in a multi-view distributed video coding system," in *Proc. Int. Conf. on Acoust., Speech and Sig. Proc. (ICASSP)*, Taipei, Taiwan, Apr. 2009.

14. C. Dikici, T. Maugey, M. Agostini, and O. Crave, "Efficient frame interpolation for wyner-ziv video coding," in *Proc. SPIE Visual Commun. and Image Processing*, San Jose, CA, USA, Jan. 2009.

15. T. Maugey, T. André, B. Pesquet-Popescu, and J. Farah, "Analysis of error propagation due to frame losses in a distributed video coding system," in *Proc. Eur. Sig. and Image Proc. Conference (EUSIPCO)*, Lausanne, Switzerland, Aug. 2008.

Appendix

Compressed sensing of multiview images based on disparity estimation methods

DISPARITY-COMPENSATED COMPRESSED-SENSING RECONSTRUCTION FOR MULTIVIEW IMAGES

Maria Trocan[†], *Thomas Maugey*[‡], *James E. Fowler*[*], *Béatrice Pesquet-Popescu*[‡]

[†]*Institut Supérieur d'Electronique de Paris*, [‡]*Télécom ParisTech*, [*]*Mississippi State University*
maria.trocan@isep.fr, {maugey, beatrice.pesquet}@telecom-paristech.fr, fowler@ece.msstate.edu

ABSTRACT

In a multiview-imaging setting, image-acquisition costs could be substantially diminished if some of the cameras operate at a reduced quality. Compressed sensing is proposed to effectuate such a reduction in image quality wherein certain images are acquired with random measurements at a reduced sampling rate via projection onto a random basis of lower dimension. To recover such projected images, compressed-sensing recovery incorporating disparity compensation is employed. Based on a recent compressed-sensing recovery algorithm for images that couples an iterative projection-based reconstruction with a smoothing step, the proposed algorithm drives image recovery using the projection-domain residual between the random measurements of the image in question and a disparity-based prediction created from adjacent, high-quality images. Experimental results reveal that the disparity-based reconstruction significantly outperforms direct reconstruction using simply the random measurements of the image alone.

Keywords— Compressed sensing, multiview, disparity compensation, directional transforms

1. INTRODUCTION

More and more applications, like 3D reconstruction, creation of virtual environments, surveillance applications, etc., require systems which capture a scene with several cameras. In these cases, the correlation between images is high because they describe the same scene. Compression, restoration, or other data processing should therefore exploit this redundancy in order to improve performance. The correlation between multiview images can be taken into account by estimating the disparity between them, which corresponds to the displacement of an object between the images and which is a quantity related to the object's depth. Since multiview technology is relatively new, the acquisition of the multiview data can be rather costly. However, the acquisition cost of multiview images could be greatly reduced if only some of the multiviews are captured at high resolution or high fidelity; the other views could possibly be acquired at a lower acquisition cost and thereby be reduced in quality. Such lower acquisition cost could be effectuated by using a compressed-sensing (CS) recovery of these latter images. CS (e.g., [1]) is a recent paradigm which allows describing a signal with a rate lower than Nyquist without any loss. This is possible under a certain hypothesis of sparsity, and is often driven by linear projection onto random basis. Such random-projection-based signal acquisition could feasibly be accomplished using a so-called single-pixel camera [2]; the corresponding reconstruction can be achieved via any one of a number of emerging schemes for CS image reconstruction (e.g., [3, 4, 5]).

In this paper, we propose to incorporate disparity compensation (DC) into the CS reconstruction of multiview images. In [4], an efficient block-based CS reconstruction of images using directional transforms was proposed. Our goal here is to improve the performance of this algorithm by considering disparity information at the reconstruction. The results that we obtain are promising and demonstrate that we can reach a recovery quality of more than 50 dB with an acquisition sampling rate divided by at least two. As previously mentioned, we anticipate that this paradigm can be useful in a multiview acquisition wherein some cameras have lower quality than others.

The remainder of the paper is organized as follows. Sec. 2 gives an overview of the CS paradigm introducing the basics for our method which is in turn presented in detail in Sec. 3. Experimental results demonstrating the efficiency of the DC scheme are presented in Sec. 4. Finally, some concluding remarks are made in Sec. 5.

2. BACKGROUND

In CS, a real-valued signal x of length N has to be recovered from M samples, where $M \ll N$ [1]. In other words, x should be reconstructed from the observations $y = \Phi x$, where y has length M, and $\Phi_{M\times N}$ is called the measurement matrix. This recovery is possible if x is sufficiently sparse in a certain space. The usual choice for the measurement basis Φ is a random matrix; in the following, we assume that Φ is orthonormal such that $\Phi\Phi^T = I$. In general, the sparsity condition for x recovery will exist with respect to some unknown transform Ψ. In this case, the key to CS reconstruction is the production of a sparse set of significant transform coefficients, $\tilde{x} = \Psi x$, and the ideal recovery procedure searches for

978-1-4244-7493-6/10/$26.00 ©2010 IEEE

the $\check{x}$ with the smallest l_0 norm consistent with the observed y. However, as this l_0 optimization is NP-complete, several alternative procedures have been proposed. For example, applying a convex relaxation to the l_0 problem results in an l_1 optimization, as exemplified by basis/matching-pursuit-based algorithms [6, 7, 8]:

$$\check{x} = \arg\min_{\check{x}} \|\check{x}\|_1 , \quad \text{such that} \;\; y = \Phi\Psi^{-1}\check{x}$$

where Ψ^{-1} represents the inverse transform. Generally, such algorithms could be implemented with linear programming.

Recently, projection-based CS-reconstruction techniques have been proposed [9]. Algorithms of this class recover $\check{x}$ by successively projecting and thresholding: the reconstruction starts from some initial approximation $\check{x}^{(0)}$, which is further refined in an iterative manner, as in the following:

$$\begin{aligned} \check{\check{x}}^{(i)} &= \check{x}^{(i)} + \frac{\Psi\Phi^T}{\lambda}(y - \Phi\Psi^{-1}\check{x}^{(i)}) \\ \check{x}^{(i+1)} &= \begin{cases} \check{\check{x}}^{(i)}, & \left|\check{\check{x}}^{(i)}\right| \geq \tau^{(i)} \\ 0, & \text{otherwise} \end{cases}, \end{aligned} \tag{1}$$

where λ is a scaling factor, and $\tau^{(i)}$ is the threshold used at the ith iteration. It is straightforward to see that this procedure is a specific instance of a projected Landweber (PL) algorithm [10].

In [3], a block-based approach of the above paradigm for the CS recovery of 2D images was proposed. In this technique, the sampling of an image is driven by random matrices applied block-by-block to the image, while the reconstruction is a variant of the PL reconstruction of (1) that incorporates a smoothing operation (e.g. Wiener filtering), ostensibly to eliminate block artifacts due to the block-based sampling. Due to its combination of block-based CS (BCS) sampling and smoothed-PL (SPL) reconstruction, this technique was denoted BCS-SPL in [4]; we adopt this same terminology here. The recovery process in BCS-SPL is iterative—the approximation of the image at iteration $i+1$, $x^{(i+1)}$, is obtained from $x^{(i)}$ as [4]:

$$\begin{aligned} &\text{function} \;\; x^{(i+1)} = \text{SPL}(x^{(i)}, y, \Phi_{\text{block}}, \Psi, \lambda) \\ &\quad \hat{x}^{(i)} = \text{Wiener}(x^{(i)}) \\ &\quad \text{for each block } j \\ &\qquad \hat{\hat{x}}_j^{(i)} = \hat{x}_j^{(i)} + \Phi_{\text{block}}^T(y - \Phi_{\text{block}}\hat{x}_j^{(i)}) \\ &\quad \check{\check{x}}^{(i)} = \Psi\hat{\hat{x}}^{(i)} \\ &\quad \check{x}^{(i)} = \text{Threshold}(\check{\check{x}}^{(i)}, \lambda) \\ &\quad \bar{x}^{(i)} = \Psi^{-1}\check{x}^{(i)} \\ &\quad \text{for each block } j \\ &\qquad x_j^{(i+1)} = \bar{x}^{(i)} + \Phi_{\text{block}}^T(y - \Phi_{\text{block}}\bar{x}_j^{(i)}) \end{aligned} \tag{2}$$

In [4], the initialization is done as $x^{(0)} = \Phi^T y$, and the reconstruction process is stopped once $\left|D^{(i+1)} - D^{(i)}\right| < 10^{-4}$, where D is defined as the mean squared error (MSE), $D^{(i)} = \frac{1}{\text{block_size}}\left\|x^{(i)} - \hat{x}^{(i-1)}\right\|_2$, between the ith image reconstruction and the first refinement step at the $(i+1)$ iteration. We note that we employ hard thresholding for the operator $\text{Threshold}(\cdot)$, where the convergence factor λ is fixed for all iterations [11] (specifically, it varies as function of the number of coefficients of Ψ from one transform to another [12]). We note also that the convergence for hard-thresholding algorithms of this nature has been proven in [13].

3. DC-BCS-SPL RECONSTRUCTION

In [4], the BCS-SPL reconstruction originating in [3] was demonstrated to provide effective reconstruction for 2D images when used with directional transforms. In the following, we propose an iterative DC algorithm for the reconstruction of multiview images; this algorithm is based on the BCS-SPL method described in the previous section and incorporates the estimation of and compensation for disparity between the multiple views. Since multiview images are strongly correlated, we exploit this correlation by deploying CS reconstruction on the DC residual. The method assumes the same setup as in [4]; that is, for the current image $\mathbf{x_d}$, which is the image to be CS-reconstructed, we have the projection/measurement matrix $\mathbf{\Phi}$; the set of observations, $\mathbf{y} = \mathbf{\Phi x_d}$; and the directional transform used in the reconstruction, $\mathbf{\Psi}$. Additionally, to adapt the BCS-SPL algorithm to the multiview scenario, we assume that we know images adjacent to $\mathbf{x_d}$; specifically, we know the closest images to the "left" and "right" of $\mathbf{x_d}$ which are $\mathbf{x_{d-1}}$ and $\mathbf{x_{d+1}}$, respectively.

The DC-BCS-SPL algorithm is partitioned into two phases. In the first phase, a predictor $\mathbf{x_p}$ for $\mathbf{x_d}$ is created by bidirectionally interpolating the closest views, $\mathbf{x_p} = \textbf{ImageInterpolation}(\mathbf{x_{d-1}}, \mathbf{x_{d+1}})$. Next, we calculate the residual $\mathbf{r}$ between the original observation $\mathbf{y}$ and the observation resulting from the projection of $\mathbf{x_p}$ using the same measurement matrix, Φ. This residual then drives the BCS-SPL reconstruction. We note that, alternatively, $\mathbf{x_p}$ could be given by the direct BCS-SPL reconstruction of the current image, i.e., $\mathbf{x_p} = \textbf{BCS-SPL}(\mathbf{y}, \mathbf{\Phi}, \mathbf{\Psi})$. However, we have found that, at low subrates (M/N small), the quality of the interpolated image is much better than that of the direct BCS-SPL reconstruction.

In the second phase, the reconstructed residual $\hat{\mathbf{r}}$ is further refined with reverse DC to obtain the final reconstruction $\hat{\mathbf{x}}_\mathbf{d}$. In the second phase, $\mathbf{DV_{d-1}}$ and $\mathbf{DV_{d+1}}$ are the left and right disparity vectors, respectively; these are obtained from disparity estimation (DE) applied to the current reconstruction, $\hat{\mathbf{x}}_\mathbf{d}$, of the current image and the left and right adjacent images. The disparity vectors then drive the DC of the current image to produce the current prediction, $\mathbf{x_p}$, and its corresponding residual, $\mathbf{r}$. We note that the second phase of the algorithm is repeated k times. The complete algorithm is

presented below:
Given $\mathbf{\Phi}$, $\mathbf{\Psi}$, and $\mathbf{y} = \mathbf{\Phi x_d}$:

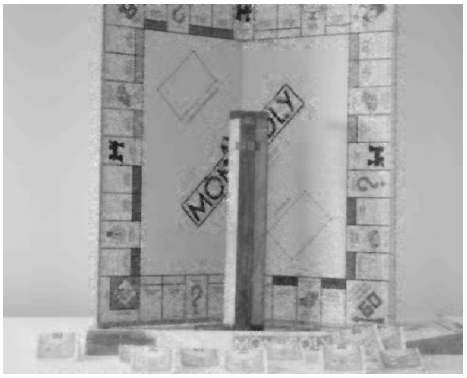
$$(\mathbf{1}) \begin{cases} \mathbf{x_p} = \textbf{ImageInterpolation}(\mathbf{x_{d-1}}, \mathbf{x_{d+1}}) \\ \mathbf{y_p} = \mathbf{\Phi x_p} \\ \mathbf{r} = \mathbf{y} - \mathbf{y_p} \\ \hat{\mathbf{r}} = \textbf{BCS-SPL}(\mathbf{r}, \mathbf{\Phi}, \mathbf{\Psi}) \\ \hat{\mathbf{x}}_\mathbf{d} = \mathbf{x_p} + \hat{\mathbf{r}} \end{cases}$$

Repeat k times:

$$(\mathbf{2}) \begin{cases} \{\mathbf{DV_{d-1}}, \mathbf{DV_{d+1}}\} = \textbf{DE}(\hat{\mathbf{x}}_\mathbf{d}, \mathbf{x_{d-1}}, \mathbf{x_{d+1}}) \\ \mathbf{x_p} = \textbf{DC}(\hat{\mathbf{x}}_\mathbf{d}, \mathbf{DV_{d-1}}, \mathbf{DV_{d+1}}) \\ \mathbf{y_p} = \mathbf{\Phi x_p} \\ \mathbf{r} = \mathbf{y} - \mathbf{y_p} \\ \hat{\mathbf{r}} = \textbf{BCS-SPL}(\mathbf{r}, \mathbf{\Phi}, \mathbf{\Psi}) \\ \hat{\mathbf{x}}_\mathbf{d} = \mathbf{x_p} + \hat{\mathbf{r}} \end{cases}$$

As illustrated in Fig. 1, the quality of DC-based reconstruction is several dBs higher than that obtained by direct BCS-SPL reconstruction. We have found this to be true regardless of the transform $\mathbf{\Psi}$ employed. Note that Fig. 1 is for a single iteration ($k = 1$) of phase 2 of the reconstruction; further improvement results from iteratively repeating phase 2. Given the quality of the reconstruction after phase 1, the predictor at each step will be obtained by DC between the current reconstructed image and its neighbors; the improvement in reconstruction quality is due to the refinement of the disparity vectors, leading to a smoother residual at each step which is much easily reconstructed by BCS-SPL.

Note that the original images ($\mathbf{x_{d-1}}$ and $\mathbf{x_{d+1}}$) are used as references for DE. This is pertinent, since the proposed algorithm serves to reduce the acquisition cost (camera quality) by at least 25% (equivalent to a subrate $M/N = 0.5$, the maximum we consider). We note also that phase 2 of the proposed algorithm converges quickly—typically, $2 \leq k \leq 5$ is sufficient for convergence in PSNR to the second decimal place.



(a)

(b)

Fig. 1. Monopoly, 512×512: BCS-SPL reconstruction using 64×64 DCT at subrate $M/N = 0.2$. (a) Direct BCS-SPL (PSNR = 29.03 dB); (b) one-step DC-BCS-SPL (PSNR = 42.70 dB).

4. EXPERIMENTAL RESULTS

In this section, we present more comprehensive experimental results, evaluating several directional transforms for both direct and DC-based CS reconstruction. Specifically, we deploy a discrete cosine (DCT), a discrete wavelet (DWT), a dual-tree discrete wavelet (DDWT) [14], and a contourlet transform (CT) [15] within the BCS-SPL framework as described in Sec. 3. We refer to the resulting implementations as *transform* for direct CS reconstruction using the transform in question, and DC-*transform* for the corresponding DC scheme using the algorithm of Sec. 3; here, *transform* $\in \{\text{DCT}, \text{DWT}, \text{DDWT}, \text{CT}\}$. In our simulations, the disparity is estimated using a full-search block-based DE algorithm, where the size of the block is 16×16, and the search area is 32×32 pixels. For BCS-SPL, we have used a 64×64 block size for the sampling and reconstruction processes. The number of decomposition levels for the tested transforms is 6. We use the BCS-SPL implementation available from its authors[1].

Figs. 2–5 present the PSNR performance for several 512×512 images from the Middlebury database[2] at several subrates, M/N. All images are rectified and the radial distortion has been removed. It should be noted that, since the quality of reconstruction can vary due to the randomness of the

[1] http://www.ece.msstate.edu/~fowler/
[2] http://cat.middlebury.edu/stereo/data.html

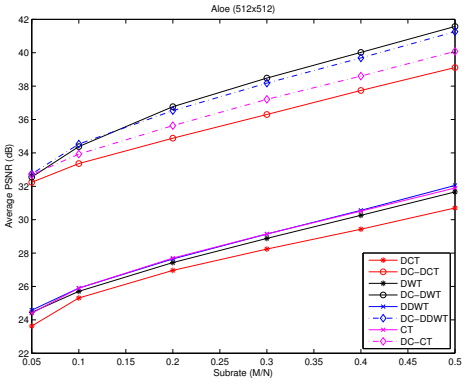


Fig. 2. Reconstruction quality (dB) for "Aloe" test image, as a function of the subrate, and for different transforms.

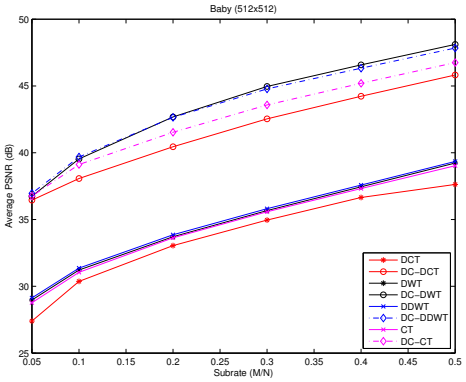


Fig. 3. Reconstruction quality (dB) for "Baby" test image, as a function of the subrate, and for different transforms.

measurement matrix $\mathbf{\Phi}$, all PSNR values in the figures are obtained by averaging 5 independent trials. It is evident that the DC-based recovery leads to higher-quality results, having an average gain of $\sim$7 dB with respect to direct BCS-SPL reconstruction. The results confirm that both direct BCS-SPL as well as DC-BCS-SPL with the DDWT achieve the best performance at both low and high subrates. Moreover, for highly textured content (e.g., the Monopoly image), the gain of the DC-based reconstruction over the direct reconstruction reaches a peak of $\sim$13 dB.

5. CONCLUSION

In this paper, we have considered the situation in which random projections coupled with CS reconstruction are used to reduce image-acquisition cost within a multiview setting. Specifically, we have assumed that an image is subject to random projections during its acquisition, and that high-quality adjacent images are available to aid its CS reconstruction. We have proposed the incorporation of DE and DC into the CS reconstruction, such that two adjacent images are used to form a prediction of the current image in between them. This predicted image is then projected using the same measurement matrix as was used to acquire the random CS projections of the current image. CS reconstruction then proceeds on the residual between the projected prediction and the projected image. Experimental results reveal a substantial increase in reconstruction quality for the DC-based algorithm as opposed to a simple, direct CS reconstruction driven by the random measurements of the image rather than the projection-domain residual.

We note that, although we have specifically considered the

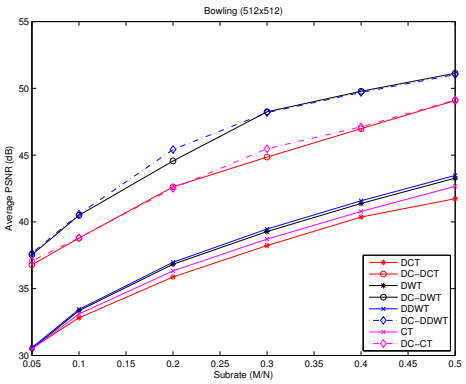


Fig. 4. Reconstruction quality (dB) for "Bowling" test image, as a function of the subrate, and for different transforms.

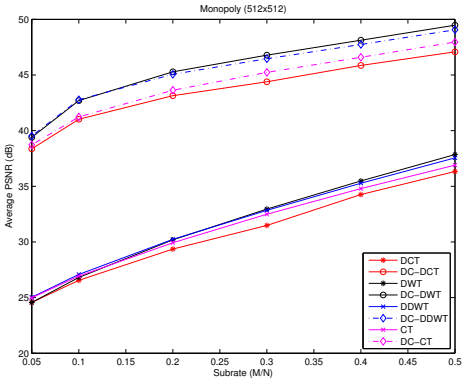


Fig. 5. Reconstruction quality (dB) for "Monopoly" test image, as a function of the subrate, and for different transforms.

multiview setting, we anticipate that the techniques presented here are also applicable to stereo images in which one image is acquired with high quality and the other is subject to CS-based random projections. In the DC-BCS-SPL algorithm we present here, one would simply modify the prediction process so as to be unidirectional rather than bidirectional.

6. REFERENCES

[1] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[2] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk, "A new compressive imaging camera architecture using optical-domain compression," in *Computational Imaging IV*, C. A. Bouman, E. L. Miller, and I. Pollak, Eds. San Jose, CA: Proc. SPIE 6065, January 2006, p. 606509.

[3] L. Gan, "Block compressed sensing of natural images," in *Proceedings of the International Conference on Digital Signal Processing*, Cardiff, UK, July 2007, pp. 403–406.

[4] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *Proceedings of the International Conference on Image Processing*, Cairo, Egypt, November 2009, pp. 3021–3024.

[5] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, August 2006.

[6] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, August 1998.

[7] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 4, pp. 586–597, December 2007.

[8] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Proceedings of the* 42th *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, California, October 2008, pp. 581–587.

[9] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, no. 49, pp. 4036–4048, 2006.

[10] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. Bristol, UK: Institute of Physics Publishing, 1998.

[11] K. K. Herrity, A. C. Gilbert, and J. A. Tropp, "Sparse approximation via iterative thresholding," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Toulouse, France, May 2006, pp. 14–19.

[12] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[13] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *The Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, December 2008.

[14] N. G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Journal of Applied Computational Harmonic Analysis*, vol. 10, pp. 234–253, May 2001.

[15] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, December 2005.

COMPRESSED SENSING OF MULTIVIEW IMAGES USING DISPARITY COMPENSATION

Maria Trocan[†], Thomas Maugey[‡], Eric W. Tramel[*], James E. Fowler[*], Béatrice Pesquet-Popescu[‡]
[†]*Institut Supérieur d'Electronique de Paris*, [‡]*Télécom ParisTech*, [*]*Mississippi State University*

ABSTRACT

Compressed sensing is applied to multiview image sets and inter-image disparity compensation is incorporated into image reconstruction in order to take advantage of the high degree of inter-image correlation common to multiview scenarios. Instead of recovering images in the set independently from one another, two neighboring images are used to calculate a prediction of a target image, and the difference between the original measurements and the compressed-sensing projection of the prediction is then reconstructed as a residual and added back to the prediction in an iterated fashion. The proposed method shows large gains in performance over straightforward, independent compressed-sensing recovery. Additionally, projection and recovery are block-based to significantly reduce computation time.

***Index Terms*—** Compressed sensing, multiview images, disparity compensation

1. INTRODUCTION

Many systems today use multiple cameras to capture information about a specified scene, such as 3D reconstruction, creation of virtual environments, and surveillance applications. Because multiview systems require multiple sensors, the cost of data acquisition is often much higher than that of traditional systems. In these multiple perspective, or multiview, situations, the correlation between images is often very high due to similar content. Compression, restoration, or other data-processing tasks can benefit greatly by exploiting this redundancy of content to improve their performance. Disparity compensation (DC) between the images within a multiview image set can be used to take advantage of this correlation.

Compressed sensing (CS) (e.g. [1]) is a recent paradigm which allows for a signal to be sampled at sub-Nyquist rates and proposes a methodology of recovery which incurs no loss. CS tells us that this is achievable under the assumption that the original signal can be described sparsely in either its ambient domain or in some other basis, $\mathbf{\Psi}$. The core of the signal-acquisition step commonly involves a projection onto a random basis, $\mathbf{\Phi}$, which must exhibit a high level of incoherence with the sparse domain [1]. Physical implementations of this methodology have been made, such as the well-known single-pixel camera [2], and many methods have been proposed for the recovery of signals acquired in this manner [3, 4, 5, 6, 7, 8].

In this paper, we propose a joint CS reconstruction algorithm for multiview image sets which takes advantage of the strong correlation between images within the set. In [4], an efficient algorithm for reconstructing randomly projected blocked images was proposed. The goal of this paper is to enhance the accuracy of this algorithm within the multiview setting through the use of inter-image DC during the reconstruction process. The results we obtain are promising and show substantial performance improvement over the straightforward, independent CS recovery of the images of the set, even at very low subsampling rates.

2. PRELIMINARIES

One of the main advantages of the CS paradigm is the very low computational burden placed on the encoding process, which requires only the projection of the signal $\mathbf{x}$, of dimensionality N, onto some measurement basis, $\mathbf{\Phi}_{N\times M}$, where $M \ll N$. The result of this computation is the M-dimensional vector of measurements, $\mathbf{y} = \mathbf{\Phi x}$. $\mathbf{\Phi}$ is often chosen to be a random matrix because it satisfies the incoherency requirements of CS reconstruction for any structured signal transform $\mathbf{\Psi}$ with a high probability. In this way, the encoder can also be said to be structure agnostic. We assume $\mathbf{\Phi}$ is also chosen to be orthonormal ($\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{I}$).

This light encoding procedure offloads most the computation of CS onto the decoder. Because the inverse of the projection $\hat{\mathbf{x}} = \mathbf{\Phi}^{-1}\mathbf{y}$ is ill-posed, we cannot directly solve the inverse problem to find the original signal from the given measurements. Instead, the CS paradigm tells us that the correct solution for $\mathbf{x}$ is the sparsest signal which lies in the set of signals that match the measurements [1]; i.e.,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\|\mathbf{\Psi x}\right\|_{\ell_0} \qquad \text{s.t.} \qquad \mathbf{y} = \mathbf{\Phi x}, \tag{1}$$

where sparsity is measured in the domain of transform $\mathbf{\Psi}$. However, this ℓ_0-constrained optimization problem is computationally infeasible due to its combinational and non-differentiable nature. Thus, a ℓ_1 convex relaxation is often applied, sacrificing accuracy but permitting the recovery to be implemented directly via linear-programming techniques (e.g., [9, 7, 8]). Further relaxations of the optimization have also been attempted, such as the mixed ℓ_1-ℓ_2 method proposed in [10]. However, all of these schemes still suffer from very long reconstructions times for N of any practical or interesting size.

Iterative thresholding algorithms have also been proposed as another class of solutions for CS recovery. The most common of these is the iterated hard thresholding (IHT) algorithm (e.g., [11, 12, 13, 14]). IHT replaces the constrained optimization formulation with an unconstrained optimization problem via a Lagrangian multiplier and further relaxes the problem by loosening the equality constraint to an ℓ_2-distance penalty,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\|\mathbf{\Psi x}\right\|_{\ell_1} + \lambda\left\|\mathbf{y} - \mathbf{\Phi x}\right\|_{\ell_2}. \tag{2}$$

Algorithms of this class recover $\hat{\mathbf{x}}$ by successive projection and thresholding operations. Given some initial approximation $\check{\mathbf{x}}^{(0)}$ to the transform coefficients $\check{\mathbf{x}} = \mathbf{\Psi x}$, the solution is calculated in the following manner:

$$\check{\tilde{\mathbf{x}}}^{(i)} = \check{\mathbf{x}}^{(i)} + \frac{1}{\gamma}\mathbf{\Psi\Phi}^T\left(\mathbf{y} - \mathbf{\Phi\Psi}^{-1}\check{\mathbf{x}}^{(i)}\right), \tag{3}$$

$$\check{\mathbf{x}}^{(i+1)} = \begin{cases} \check{\tilde{\mathbf{x}}}^{(i)}, & \left|\check{\tilde{\mathbf{x}}}^{(i)}\right| \geq \tau^{(i)}, \\ 0 & \text{else}, \end{cases} \tag{4}$$

where γ is a scaling factor, and $\tau^{(i)}$ is the threshold used at the ith iteration. Further observation of this process shows us that this procedure is actually a specific instance of a projected Landweber (PL) algorithm [15]. We note that convergence of IHT has been shown in [5].

IHT recovery improves reconstruction speed by at least an order of magnitude and maintains a high degree of accuracy. Reconstruction time can be further reduced by implementing a block-based measurement and recovery procedure, as proposed in [3]. In this technique, $\mathbf{\Phi}$ is applied on a block-by-block basis, while the reconstruction step incorporates a smoothing operation (such as Weiner filtering) into the IHT. By employing blocking, the results in [3] show a reduction of computation time by four orders of magnitude for comparable accuracy versus linear programming approaches. In [4], this method is referred to as block CS and smoothed PL (BCS-SPL) and is extended via the use of directional transforms. The algorithm in [4] is given as

$$
\begin{aligned}
&\texttt{function}\ \mathbf{x}^{(i+1)} = \mathrm{SPL}(\mathbf{x}^{(i)}, \mathbf{y}, \mathbf{\Phi}_{\text{block}}, \mathbf{\Psi}, \lambda)\\
&\quad \hat{\mathbf{x}}^{(i)} = \mathrm{Wiener}(\mathbf{x}^{(i)})\\
&\quad \texttt{for each block } j\\
&\quad\quad \hat{\hat{\mathbf{x}}}_j^{(i)} = \hat{\mathbf{x}}_j^{(i)} + \mathbf{\Phi}_{\text{block}}^T(\mathbf{y} - \mathbf{\Phi}_{\text{block}}\hat{\mathbf{x}}_j^{(i)})\\
&\quad \check{\mathbf{x}}^{(i)} = \mathbf{\Psi}\hat{\hat{\mathbf{x}}}^{(i)}\\
&\quad \bar{\mathbf{x}}^{(i)} = \mathrm{Threshold}(\check{\mathbf{x}}^{(i)}, \lambda)\\
&\quad \bar{\mathbf{x}}^{(i)} = \mathbf{\Psi}^{-1}\bar{\mathbf{x}}^{(i)}\\
&\quad \texttt{for each block } j\\
&\quad\quad \mathbf{x}_j^{(i+1)} = \bar{\mathbf{x}}_j^{(i)} + \mathbf{\Phi}_{\text{block}}^T(\mathbf{y} - \mathbf{\Phi}_{\text{block}}\bar{\mathbf{x}}_j^{(i)})
\end{aligned}
$$

Here, $\mathbf{x}^{(0)} = \mathbf{\Phi}^T\mathbf{y}$. The method uses hard thresholding with a fixed convergence factor λ for all iterations [6], and can be calculated as a function of the number of coefficients used in $\mathbf{\Psi}$ [16].

3. DC-BCS-SPL

In [4], BCS-SPL was shown to be both more computationally efficient and to provide more accurate reconstructions than other recovery techniques, especially when using directional transforms as the sparse basis. We now propose a method which incorporates disparity estimation and compensation as side information into this competitive recovery algorithm with the goal of improving recovery accuracy when used within the multiview setting. We exploit the strong correlations between multiview images by reconstructing the residual between images and their disparity-compensated predictions as a means for refining the accuracy of direct BCS-SPL reconstruction. Our method requires no additional information from the encoder, simply the typical CS formulation—namely, the measurement matrix, $\mathbf{\Phi}_d$; a set of measurements, $\mathbf{y} = \mathbf{\Phi}_d\mathbf{x}_d$; and the sparsity basis, $\mathbf{\Psi}$. We refer to this proposed method as disparity-compensated BCS-SPL (DC-BCS-SPL).

The DC-BCS-SPL algorithm, depicted in Fig. 1, is partitioned into two phases. In the first phase, a prediction of the current image, $\mathbf{x}_d$, is created by bidirectionally interpolating the BCS-SPL reconstructions of the two nearest views (the left and right neighbors), i.e. $\mathbf{x}_p = \mathrm{ImageInterpolation}(\hat{\mathbf{x}}_{d-1}, \hat{\mathbf{x}}_{d+1})$. Next, the residual, $\mathbf{r}$ is calculated by taking the difference between the given measurements, $\mathbf{y}_d$, and the projection of $\mathbf{x}_p$ onto the measurement basis, $\mathbf{y}_p = \mathbf{\Phi}_d\mathbf{x}_p$. This residual, $\mathbf{r} = \mathbf{y}_p - \mathbf{y}$, is then reconstructed using BCS-SPL and added back to $\mathbf{x}_p$ to obtain the reconstruction $\hat{\mathbf{x}}_d$.

In the second phase, the reconstruction obtained from the first phase is used to refine the prediction, $\mathbf{x}_p$. Disparity estimation is used to find two sets of disparity vectors, $\mathbf{DV}_{d-1}$ and $\mathbf{DV}_{d+1}$, between $\hat{\mathbf{x}}_d$ and the reconstructions of its neighbor images. The disparity vectors are then used to produce two disparity-compensated predictions of $\hat{\mathbf{x}}_d$ which are averaged together to produce a single prediction. This prediction will serve as the $\mathbf{x}_p$ for the next reconstruction. This process is repeated k times.

The iterative process improves the quality of the final reconstruction because the use of DC allows us to make a better prediction of the image, which leads to smoother and more easily reconstructed residuals, which then allow us to make more accurate predictions, and so on. DC-BCS-SPL converges quickly—typically iterating for $2 \leq k \leq 5$ is sufficient.

4. EXPERIMENTAL RESULTS

In order to observe the effectiveness of the DC-BCS-SPL recovery, we evaluate the performance of the proposed method against that of the direct-recovery approach, i.e., BCS-SPL used to reconstruct the frame independently of its neighbors. We use several transforms, specifically a DCT, DWT, complex dual-tree DWT (DDWT), and contourlet transform (CT). In our results, we refer to the implementations of the direct approach simply by the name of the used transform, and DC-*transform* is used to refer to the implementations of DC-BCS-SPL using the named transform. In our simulations, disparity vectors are calculated using a full block-based search with integer-pixel accuracy, a block size of 16×16, and a search window of 32×32. It is conceivable that the performance of DC-BCS-SPL could be increased with more sophisticated disparity-vector estimation. For DC-BCS-SPL, we consider two measurement block sizes, 32×32 and 64×64, and the wavelet based transforms are computed to 5 and 6 levels of decomposition, respectively, for these block sizes. Additionally, all images within the measured multiview set are projected using the same subrate.

Tables 1 and 2 present the performance, in PSNR, for several 512×512 images from the Middlebury multiview database[1] at several subrates, M/N, and for the two measurement block sizes considered. All images are rectified, and any radial distortion is removed. It should be noted that, due to the variation in quality that can result from differences in random measurement matrices, all PSNR values represent an average of 5 independent trials.

As illustrated in Fig. 2, the quality of DC-BCS-SPL is overall $\sim$2 dBs higher than the PSNR performance obtained by using direct BCS-SPL under the same conditions. We have found this performance gain to be true regardless of the sparsity basis, $\mathbf{\Psi}$, used. Note that results in Fig. 2 are calculated by using a single iteration ($k = 1$) of reconstruction. Increasing the number of iterations shows further performance gains.

The DC-BCS-SPL method shows a performance improvement of $\sim$1 dB to $\sim$3 dB for lower to higher subrates in comparison to direct BCS-SPL. Of the transforms used, the DDWT gave the best performance for both direct and DC BCS-SPL. Additionally, for images with high variation or texture (such as the "Monopoly" multiview image set), the performance gain of the DC method over direct BCS-SPL is even more pronounced, peaking at $\sim$4.5 dB. It should also be noted that low-variation images benefited from larger measurement block sizes, as can be seen for the "Plastic" multiview image set which shows a performance gain of $\sim$1.5 dBs when 64×64 blocks are used instead of 32×32 blocks.

5. CONCLUSIONS

In this paper, we proposed a new method for the CS recovery of multiview images which takes advantage of the high degree of inter-frame correlation which is characteristic of the multiview application. We included side information in the form of disparity

[1] `http://cat.middlebury.edu/stereo/data.html`

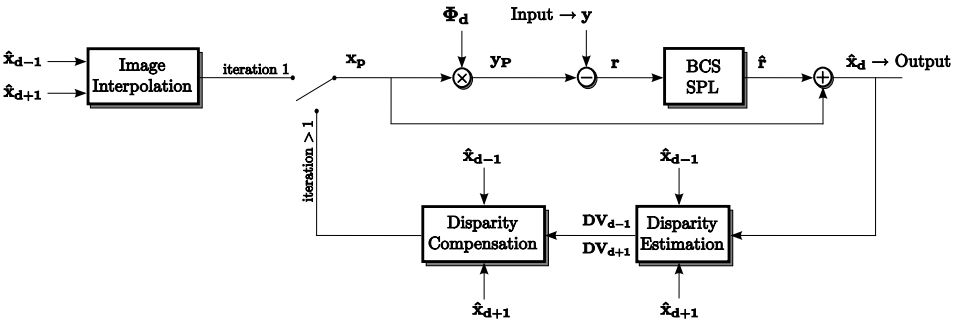


Figure 1: The DC-BCS-SPL reconstruction algorithm.

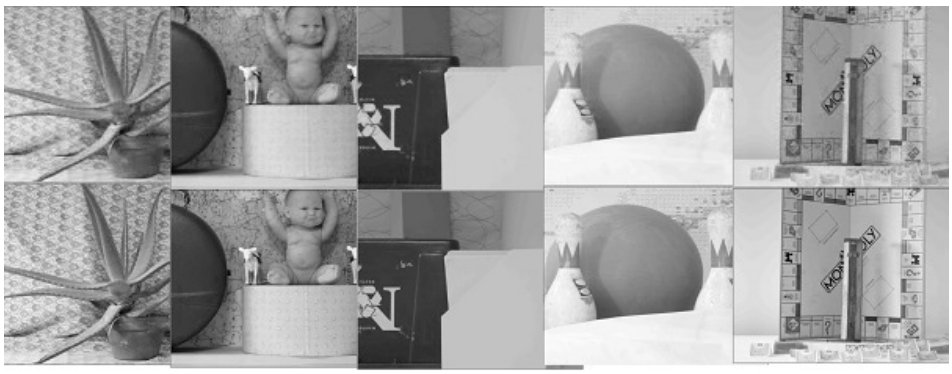


Figure 2: Images from the five multiview sets (left to right: Aloe, Baby, Plastic, Bowling, and Monopoly) reconstructed using the given experimental framework: the first row using direct BCS-SPL, the second row using DC-BCS-SPL

estimation and compensation and using the technique of reconstructing a residual rather than an image, and we incorporated this information into the CS-recovery framework. Experimental results displayed an increase in performance when using this extra information in comparison to recoveries which merely reconstruct each image independently from one another.

6. REFERENCES

[1] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[2] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, March 2008.

[3] L. Gan, "Block compressed sensing of natural images," in *Proceedings of the International Conference on Digital Signal Processing*, Cardiff, UK, July 2007, pp. 403–406.

[4] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *Proceedings of the International Conference on Image Processing*, Cairo, Egypt, November 2009, pp. 3021–3024.

[5] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *The Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, December 2008.

[6] K. K. Herrity, A. C. Gilbert, and J. A. Tropp, "Sparse approximation via iterative thresholding," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Toulouse, France, May 2006, pp. 14–19.

[7] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal*

Table 1: PSNR performance for Aloe, Baby, Plastic, Bowling, Monopoly (512×512, Middlebury database): 32×32 blocksize for BCS-SPL.

Aloe					
Subrate/PSNR (dB)	0.1	0.2	0.3	0.4	0.5
DCT	25.24	26.95	28.23	29.44	30.69
DC-DCT	25.67	28.16	30.00	31.77	33.59
DWT	25.70	27.44	28.91	30.31	31.67
DC-DWT	26.34	29.08	31.16	33.04	34.89
DDWT	25.88	27.68	29.17	30.61	32.07
DC-DDWT	**26.61**	**29.34**	**31.50**	**33.47**	**35.43**
CT	25.88	27.75	29.19	30.56	31.93
DC-CT	26.55	29.27	31.27	33.10	34.91
Baby					
Subrate/PSNR (dB)	0.1	0.2	0.3	0.4	0.5
DCT	30.51	33.16	35.11	36.86	38.60
DC-DCT	31.34	34.65	37.00	39.15	41.30
DWT	30.77	33.61	35.64	37.45	39.25
DC-DWT	31.49	35.53	38.07	40.32	42.52
DDWT	31.00	33.78	35.79	37.60	39.37
DC-DDWT	**32.13**	**35.77**	**38.26**	**40.56**	**42.72**
CT	30.84	33.62	35.63	37.42	39.16
DC-CT	32.12	35.48	37.78	39.86	41.89
Plastic					
Subrate/PSNR (dB)	0.1	0.2	0.3	0.4	0.5
DCT	31.98	35.94	39.12	41.76	44.03
DC-DCT	**32.68**	36.69	40.39	44.26	47.32
DWT	31.58	36.04	39.58	42.64	45.31
DC-DWT	31.57	35.16	38.66	44.32	47.79
DDWT	31.72	36.28	39.88	43.02	45.84
DC-DDWT	31.38	35.24	39.13	44.04	**48.97**
CT	32.03	36.35	39.39	42.05	44.48
DC-CT	31.99	**37.04**	**41.51**	**44.64**	47.07
Bowling					
Subrate/PSNR (dB)	0.1	0.2	0.3	0.4	0.5
DCT	32.41	35.44	37.65	39.79	41.76
DC-DCT	33.33	37.00	39.80	42.10	44.54
DWT	32.60	35.96	38.42	40.61	42.64
DC-DWT	33.36	37.61	40.96	43.46	45.85
DDWT	32.70	36.08	38.61	40.87	42.94
DC-DDWT	33.66	**38.10**	**41.54**	**44.07**	**46.56**
CT	32.55	35.76	38.06	40.20	42.15
DC-CT	**33.74**	37.48	40.31	42.54	44.65
Monopoly					
Subrate/PSNR (dB)	0.1	0.2	0.3	0.4	0.5
DCT	26.34	28.74	31.55	33.78	36.00
DC-DCT	27.95	32.03	34.86	37.82	40.35
DWT	26.15	29.26	31.89	34.34	36.76
DC-DWT	27.29	32.39	36.05	39.20	41.98
DDWT	26.23	29.49	32.28	34.79	37.19
DC-DDWT	27.48	32.82	**36.58**	**39.55**	**42.18**
CT	26.73	29.58	32.10	34.42	36.62
DC-CT	**28.73**	**33.06**	35.99	38.58	40.96

Table 2: PSNR performance for Aloe, Baby, Plastic, Bowling, Monopoly (512×512, Middlebury database): 64×64 blocksize for BCS-SPL.

Aloe						
Subrate/PSNR (dB)	0.05	0.1	0.2	0.3	0.4	0.5
DCT	23.63	25.31	26.96	28.24	29.43	30.70
DC-DCT	24.01	25.81	28.13	29.94	31.66	33.45
DWT	24.45	25.71	27.43	28.88	30.26	31.66
DC-DWT	24.78	26.48	29.10	31.15	33.02	34.90
DDWT	24.58	25.90	27.65	29.14	30.56	32.05
DC-DDWT	**24.89**	**26.66**	**29.33**	**31.48**	**33.45**	**35.44**
CT	24.40	25.90	27.70	29.15	30.51	31.90
DC-CT	24.73	26.61	29.25	31.28	33.10	34.94
Baby						
Subrate/PSNR (dB)	0.05	0.1	0.2	0.3	0.4	0.5
DCT	27.40	30.37	33.05	34.96	36.65	37.63
DC-DCT	28.59	31.34	34.49	36.77	38.71	40.67
DWT	28.97	31.22	33.71	35.67	37.45	39.23
DC-DWT	29.34	32.08	35.61	38.08	40.28	42.47
DDWT	29.13	31.36	33.86	35.81	37.58	39.35
DC-DDWT	29.78	32.31	**35.67**	**38.25**	**40.50**	**42.69**
CT	28.77	31.05	33.63	35.58	37.32	39.03
DC-CT	**29.84**	**32.37**	35.57	37.91	39.99	42.04
Plastic						
Subrate/PSNR (dB)	0.05	0.1	0.2	0.3	0.4	0.5
DCT	30.17	32.72	36.72	38.93	41.09	44.08
DC-DCT	**30.73**	**33.92**	38.89	42.22	44.43	47.73
DWT	28.96	32.66	37.27	40.97	44.10	46.76
DC-DWT	29.70	33.50	39.22	45.10	48.62	50.95
DDWT	29.39	32.84	37.54	41.28	44.47	47.14
DC-DDWT	29.91	33.54	**40.23**	**45.96**	**49.02**	**51.27**
CT	29.87	33.07	37.12	40.19	42.81	45.18
DC-CT	30.01	33.52	39.28	43.35	46.40	49.25
Bowling						
Subrate/PSNR (dB)	0.05	0.1	0.2	0.3	0.4	0.5
DCT	30.55	32.82	35.88	38.23	40.36	41.74
DC-DCT	31.32	34.11	37.65	40.34	42.81	45.09
DWT	30.49	33.36	36.83	39.28	41.38	43.28
DC-DWT	31.26	34.85	39.29	42.24	44.66	46.77
DDWT	30.59	33.45	36.98	39.47	41.58	43.48
DC-DDWT	31.58	**35.21**	**39.71**	**42.60**	**44.94**	**47.06**
CT	30.47	33.11	36.33	38.71	40.78	42.66
DC-CT	**31.59**	34.67	38.59	41.26	43.49	45.72
Monopoly						
Subrate/PSNR (dB)	0.05	0.1	0.2	0.3	0.4	0.5
DCT	24.57	26.55	29.36	31.48	34.26	36.33
DC-DCT	25.42	28.02	32.03	35.13	37.89	39.89
DWT	24.56	26.81	30.19	32.95	35.48	37.85
DC-DWT	24.97	28.25	33.56	**37.13**	**40.00**	**42.64**
DDWT	25.03	27.08	30.23	32.84	35.27	37.55
DC-DDWT	25.24	28.63	**33.58**	36.78	39.50	42.11
CT	24.98	26.93	29.93	32.49	34.79	36.90
DC-CT	**25.97**	**29.09**	33.29	36.30	38.86	41.23

on Selected Areas in Communications, vol. 1, no. 4, pp. 586–597, December 2007.

[8] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Proceedings of the 42th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, California, October 2008, pp. 581–587.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, August 1998.

[10] X. Chen and P. Frossard, "Joint reconstruction of compressed multi-view images," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, April 2009, pp. 1005–1008.

[11] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, November 2009.

[12] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, December 2007.

[13] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, September 2006.

[14] M. Fornasier and H. Rauhut, "Iterative thresholding algorithms," *Applied and Computational Harmonic Analysis*, vol. 25, no. 2, pp. 187–208, September 2008.

[15] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. Bristol, UK: Institute of Physics Publishing, 1998.

[16] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

Multistage Compressed-Sensing Reconstruction of Multiview Images

Maria Trocan *, Thomas Maugey +, Eric W. Tramel #, James E. Fowler #, Béatrice Pesquet-Popescu +

* *Institut Supérieur d'Électronique de Paris, France*
*maria.trocan@isep.fr

+ *Télécom ParisTech, France*

Mississippi State University, USA

***Abstract*—Compressed sensing is applied to multiview image sets and the high degree of correlation between views is exploited to enhance recovery performance over straightforward independent view recovery. This gain in performance is obtained by recovering the difference between a set of acquired measurements and the projection of a prediction of the signal they represent. The recovered difference is then added back to the prediction, and the prediction and recovery procedure is repeated in an iterated fashion for each of the views in the multiview image set. The recovered multiview image set is then used as an initialization to repeat the entire process again to form a multistage refinement. Experimental results reveal substantial performance gains from the multistage reconstruction.**

I. Introduction

Many modern applications, such as 3D reconstruction, creation of virtual environments, surveillance systems, and more, require several cameras to record a scene concurrently from different perspectives. In these cases, there is a large amount correlation between the images representing each viewpoint. Compression, restoration, or other data-processing techniques can make use of this information redundancy to enhance their performance or robustness. Disparity compensation (DC) is commonly used to exploit this redundancy by making a prediction of a current view from other views in the image set. In the case of compression, a DC prediction can be used to calculate a residual between the prediction and the original image. The residual image obtained in this manner is often much more amenable to compression than the original image.

Because multiview data acquisition requires many sensors operating concurrently, the volume of data to be either stored locally or transmitted remotely can be prohibitive in some applications. It is anticipated that such applications can benefit from compressed sensing (CS), a new paradigm which allows signals to be sampled at sub-Nyquist rates and, under certain conditions of sparsity and incoherence [1], be recovered with negligible loss. One common method of CS-based signal acquisition uses a linear projection onto a random basis, a scenario that has been shown to be physically realizable with a single-pixel camera [2]. Recovery of signals sampled in this manner can be achieved via any one of the many proposed CS reconstruction schemes (e.g., [3]).

In this paper, we propose a joint CS reconstruction of multiview image sets by utilizing DC to form predictions which serve as a form of side information to the image reconstruction algorithm. We use the efficient block-based method proposed in [4] as our image-recovery procedure. Experimental results indicate that the proposed method shows promising performance and demonstrates high-quality reconstruction even at very low subsampling rates. We note that a preliminary system we described in [5, 6] used an approach similar to that considered here; however, the system of [5, 6] employed a simpler, two-stage recontruction. In contrast, the system we propose here adds one or more refinement stages to produce a multistage reconstruction exhibiting substantial improvement in performance over the system of [5, 6].

II. Background

One of the main advantages of the CS paradigm is the very low computational burden placed on the encoding process, which requires only the projection of the signal $\mathbf{x}$, of dimensionality N, onto some measurement basis, $\mathbf{\Phi}_{N\times M}$, where $M \ll N$. The result of this computation is the M-dimensional vector of measurements, $\mathbf{y} = \mathbf{\Phi x}$. $\mathbf{\Phi}$ is often chosen to be a random matrix because it satisfies the incoherency requirements of CS reconstruction for any structured signal transform $\mathbf{\Psi}$ with a high probability. In this way, the encoder can also be said to be structure agnostic. We assume $\mathbf{\Phi}$ is also chosen to be orthonormal ($\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{I}$). We define the subsampling rate, or subrate, of the CS scheme as M/N.

This light encoding procedure offloads most the computation of CS onto the decoder. Because the inverse of the projection $\mathbf{\Phi}$ is ill-posed, we cannot directly solve the inverse problem to find the original signal from the given measurements. Instead, the CS paradigm tells us that the correct solution for $\mathbf{x}$ is the sparsest signal which lies in the set of signals that match the measurements [1]; i.e.,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\|\mathbf{\Psi x}\right\|_{\ell_0} \qquad \text{s.t.} \qquad \mathbf{y} = \mathbf{\Phi x}, \tag{1}$$

where sparsity is measured in the domain of transform $\mathbf{\Psi}$. However, this ℓ_0-constrained optimization problem is computationally infeasible due to its combinational and non-differentiable nature. Thus, a ℓ_1 convex relaxation is often applied, sacrificing accuracy but permitting the recovery to be implemented directly via linear-programming techniques (e.g., [7–9]). Further relaxations of the optimization have also

been attempted, such as the mixed ℓ_1-ℓ_2 method proposed in [10]. However, all of these schemes still suffer from very long reconstructions times for N of any practical or interesting size.

Iterative thresholding algorithms have also been proposed as another class of solutions for CS recovery. The most common of these is the iterated hard thresholding (IHT) algorithm (e.g., [11–13]). IHT replaces the constrained optimization formulation with an unconstrained optimization problem via a Lagrangian multiplier and further relaxes the problem by loosening the equality constraint to an ℓ_2-distance penalty,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \ \left\|\mathbf{\Psi x}\right\|_{\ell_1} + \lambda\left\|\mathbf{y} - \mathbf{\Phi x}\right\|_{\ell_2}. \tag{2}$$

Algorithms of this class recover $\hat{\mathbf{x}}$ by successive projection and thresholding operations. Given some initial approximation $\bar{\mathbf{x}}^{(0)}$ to the transform coefficients $\check{\mathbf{x}} = \mathbf{\Psi x}$, the solution is calculated in the following manner:

$$\check{\bar{\mathbf{x}}}^{(i)} = \check{\mathbf{x}}^{(i)} + \frac{1}{\gamma}\mathbf{\Psi\Phi}^T\left(\mathbf{y} - \mathbf{\Phi\Psi}^{-1}\check{\mathbf{x}}^{(i)}\right), \tag{3}$$

$$\check{\mathbf{x}}^{(i+1)} = \begin{cases} \check{\bar{\mathbf{x}}}^{(i)}, & \left|\check{\bar{\mathbf{x}}}^{(i)}\right| \geq \tau^{(i)}, \\ 0 & \text{else}, \end{cases} \tag{4}$$

where γ is a scaling factor, and $\tau^{(i)}$ is the threshold used at the ith iteration. Further observation of this process shows us that this procedure is actually a specific instance of a projected Landweber (PL) algorithm [14]. We note that convergence of IHT has been shown in [11].

IHT recovery improves reconstruction speed by at least an order of magnitude and maintains a high degree of accuracy. Reconstruction time can be further reduced by implementing a block-based measurement and recovery procedure, as proposed in [3]. In this technique, $\mathbf{\Phi}$ is applied on a block-by-block basis, while the reconstruction step incorporates a smoothing operation (such as Wiener filtering) into the IHT. By employing blocking, the results in [3] show a reduction of computation time by four orders of magnitude for comparable accuracy versus linear-programming approaches. In [4], this method is referred to as block CS and smoothed PL (BCS-SPL) and is extended via the use of directional transforms. The algorithm in [4] is given as

$$\begin{aligned}
&\texttt{function } \mathbf{x}^{(i+1)} = \mathrm{SPL}(\mathbf{x}^{(i)}, \mathbf{y}, \mathbf{\Phi}_{\text{block}}, \mathbf{\Psi}, \lambda) \\
&\quad \hat{\mathbf{x}}^{(i)} = \mathrm{Wiener}(\mathbf{x}^{(i)}) \\
&\quad \texttt{for each block } j \\
&\qquad \hat{\mathbf{x}}_j^{(i)} = \hat{\mathbf{x}}_j^{(i)} + \mathbf{\Phi}_{\text{block}}^T(\mathbf{y} - \mathbf{\Phi}_{\text{block}}\hat{\mathbf{x}}_j^{(i)}) \\
&\quad \check{\mathbf{x}}^{(i)} = \mathbf{\Psi}\hat{\mathbf{x}}^{(i)} \\
&\quad \check{\bar{\mathbf{x}}}^{(i)} = \mathrm{Threshold}(\check{\mathbf{x}}^{(i)}, \lambda) \\
&\quad \bar{\mathbf{x}}^{(i)} = \mathbf{\Psi}^{-1}\check{\bar{\mathbf{x}}}^{(i)} \\
&\quad \texttt{for each block } j \\
&\qquad \mathbf{x}_j^{(i+1)} = \bar{\mathbf{x}}_j^{(i)} + \mathbf{\Phi}_{\text{block}}^T(\mathbf{y} - \mathbf{\Phi}_{\text{block}}\bar{\mathbf{x}}_j^{(i)})
\end{aligned}$$

Here, $\mathbf{x}^{(0)} = \mathbf{\Phi}^T\mathbf{y}$. The method uses hard thresholding with a fixed convergence factor λ for all iterations [13], and can be calculated as a function of the number of coefficients used in $\mathbf{\Psi}$ [15].

III. Disparity-Compensated CS Reconstruction

We propose an iterative disparity-compensated algorithm for the reconstruction of multiview images using BCS-SPL. Because multiview images are strongly correlated, we can exploit this redundancy and consider only the DC residual for CS reconstruction. The given method assumes the same context as [4]. Each image in the multiview set, $\mathbf{x}_d$, is acquired using a measurement matrix, $\mathbf{\Phi}_d$, and the decoder is given only the set of observations $\mathbf{y}_d = \mathbf{\Phi}_d\mathbf{x}_d$ along with each $\mathbf{\Phi}_d$ used. The decoder makes a blind decision on the sparse basis, $\mathbf{\Psi}$, to use.

The algorithm is partitioned into three stages, as can be seen in the block diagram in Fig. 1. In the first, or initial, stage, each image in the multiview set is reconstructed individually from the received set of measurements using BCS-SPL. In the second stage (the "basic" stage), for each image $\mathbf{x}_d$, a prediction, $\mathbf{x}_p$, is created by bidirectionally interpolating the BCS-SPL reconstructions of the closest views, $\mathbf{x}_p = \mathrm{ImageInterpolation}(\hat{\mathbf{x}}_{d-1}, \hat{\mathbf{x}}_{d+1})$. Alternatively, the direct reconstruction of the view as obtained from BCS-SPL could be used as the initial prediction. However, we have found that at low subrates, the quality of the final reconstruction is much better when using an interpolation as the initial prediction. Next, we compute the residual $\mathbf{r}$ between the measurements and the projection of $\mathbf{x}_p$ by $\mathbf{\Phi}_d$. This residual in the measurement domain is then reconstructed using BCS-SPL and added back to the prediction to generate a reconstruction, $\hat{\mathbf{x}}_d$.

$\hat{\mathbf{x}}_d$ is further refined in the basic stage by calculating a set of disparity vectors, $\mathbf{DV}_{d-1}$ and $\mathbf{DV}_{d+1}$ (the right and left disparity vectors, respectively), via disparity estimation using the reconstructions of the neighboring right and left views from the first stage. These disparity vectors then drive the DC to form a prediction of the current view from these neighboring views. This prediction is substituted for $\mathbf{x}_p$, and the procedure is repeated. This procedure improves the quality of $\hat{\mathbf{x}}_d$ at each iteration by refining the disparity vectors at each step, producing better predictions and therefore producing smoother residuals which are more accurately recovered, leading finally to a more accurate $\hat{\mathbf{x}}_d$. For our implementation, we iterated three times.

Subsequently, one or more bootstrapping stages are performed. A bootstrapping-refinement stage of the algorithm is simply the repetition of the basic stage as described above with the results from the second stage substituted for the references used to drive the DC-CS reconstruction. The stages could conceivably be repeated until there is no significant difference between consecutive passes; however, in our experimental framework described in the next section, we consider only one refinement stage in order to minimize the overall computational complexity of the reconstruction.

We note that, for each view, a different random measurement matrix is used, and the information retained in the different projections has a high probability of being comple-

mentary. Knowing that each view is highly correlated, the performance gains from the refinement iterations is also due to complementary, highly correlated information along the disparity axis. Finally, we note that the system considered in [5, 6] used only the initial and basic stages described here; experimental results below, however, demonstrate substantial performance improvement due to the bootstrapping/refinement stage in the multistage reconstruction.

IV. Experimental results

In our experiments, we used the dual-tree discrete wavelet transform (DDWT) transform [16] as the sparse representation basis, $\mathbf{\Psi}$. The performance characteristics of the DDWT within the CS framework has been investigated in [4]. In our results, the direct reconstruction using BCS-SPL (i.e., the output of the initial stage of the algorithm) is referred to as "DDWT." On the other hand, "DC-DDWT" refers to the results obtained after the basic stage of the proposed method, while "MS-DC-DDWT" refers to the results obtained using a third, bootstrapping stage. The DC prediction for each view is calculated using a block size of 16×16 pixels with a search window of 32×32 pixels. For BCS-SPL, a block size of 64×64 pixels is used as well as 6 levels of DDWT decomposition. All views are acquired with the same subrate, M/N.

Figs. 2–6 show the PSNR performance obtained for several 512×512 images from the Middlebury stereo-image database[1] over the subrate used. All images are rectified and corrected for radial distortion. Because the measurement basis is random, all PSNR results are averaged over five independent trials.

As seen in the figures, the bootstrapping stage yields high-quality results, showing a gain of approximately 1.5 dB to 0.75 dB for high and low subrates, respectively, as compared to using only two stages. For highly textured images (e.g., "Monopoly," "Aloe"), the last stage greatly improves the final reconstruction quality; for smooth images (e.g., "Plastic"), the gains are more nominal.

V. Conclusion

In this paper, we proposed a new method of CS reconstruction for highly correlated multiview image sets. By way of a multistage refinement procedure, we use the performance gains obtained via residual recovery to promote even better performance. The residual recovery was implemented by using DC to create image predictions which were projected into the measurement domain and subtracted from the measurements of the original image. These residuals were then added back to the predictions to get final reconstructions more accurate than direct reconstruction. Repeating the procedure was shown in our results to garner even better PSNR performance.

[1] `http://cat.middlebury.edu/stereo/data.html`

References

[1] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[2] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, March 2008.

[3] L. Gan, "Block compressed sensing of natural images," in *Proceedings of the International Conference on Digital Signal Processing*, Cardiff, UK, July 2007, pp. 403–406.

[4] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *Proceedings of the International Conference on Image Processing*, Cairo, Egypt, November 2009, pp. 3021–3024.

[5] M. Trocan, T. Maugey, J. E. Fowler, and B. Pesquet-Popescu, "Disparity-compensated compressed-sensing reconstruction for multiview images," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, Singapore, July 2010, to appear.

[6] M. Trocan, T. Maugey, E. W. Tramel, J. E. Fowler, and B. Pesquet-Popescu, "Compressed sensing of multiview images using disparity compensation," in *Proceedings of the International Conference on Image Processing*, Hong Kong, September 2010, submitted.

[7] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 4, pp. 586–597, December 2007.

[8] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Proceedings of the 42th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, California, October 2008, pp. 581–587.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, August 1998.

[10] X. Chen and P. Frossard, "Joint reconstruction of compressed multi-view images," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, April 2009, pp. 1005–1008.

[11] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *The Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, December 2008.

[12] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, September 2006.

[13] K. K. Herrity, A. C. Gilbert, and J. A. Tropp, "Sparse approximation via iterative thresholding," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Toulouse, France, May 2006, pp. 14–19.

[14] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. Bristol, UK: Institute of Physics Publishing, 1998.

[15] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[16] N. G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Journal of Applied Computational Harmonic Analysis*, vol. 10, pp. 234–253, May 2001.

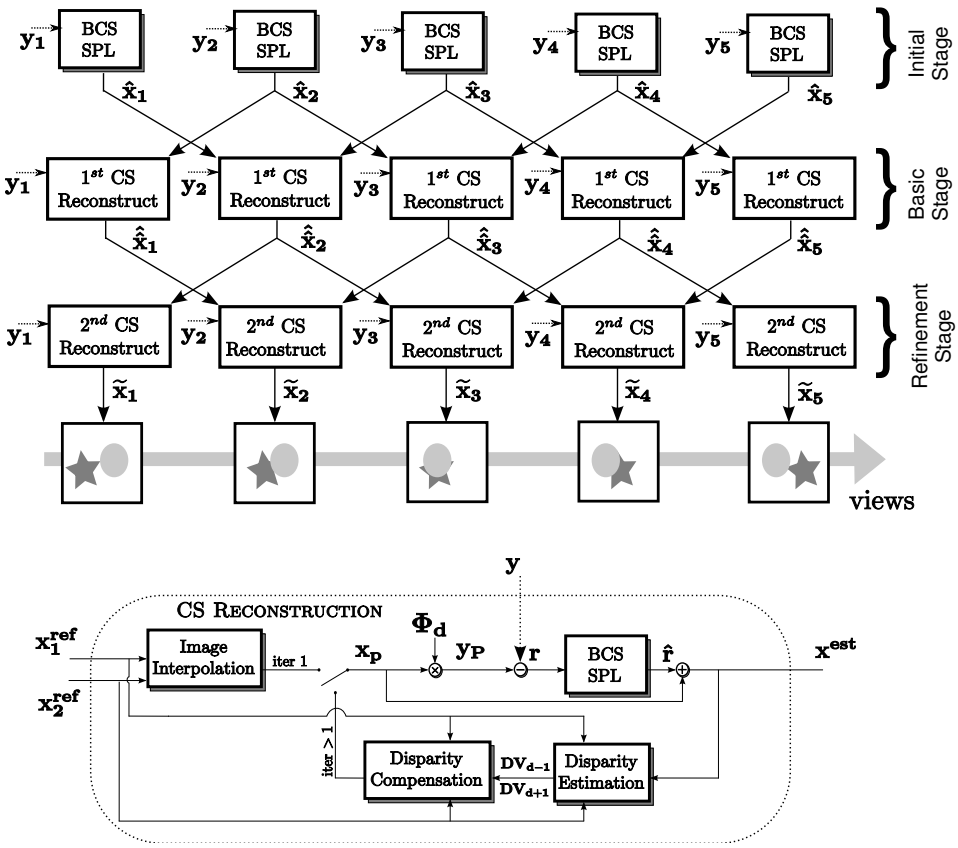


Fig. 1. The multistage DC-based reconstruction algorithm.

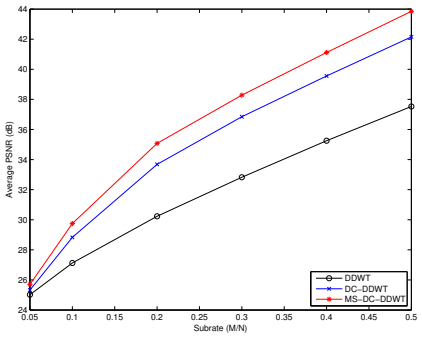


Fig. 2. Reconstruction quality for “Monopoly” as a function of subrate.

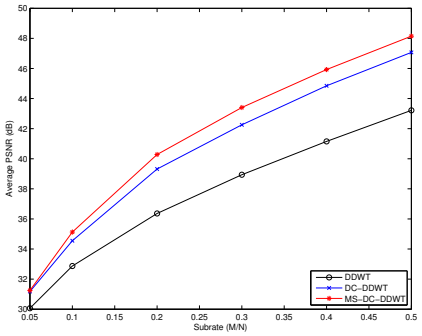


Fig. 3. Reconstruction quality for “Bowling” as a function of subrate.

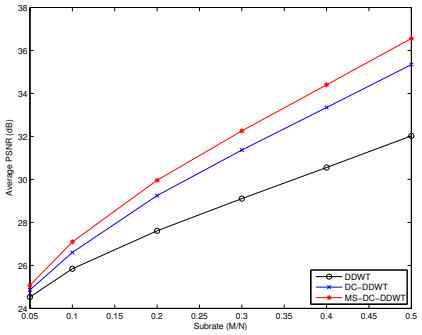


Fig. 4. Reconstruction quality for “Aloe” as a function of subrate.

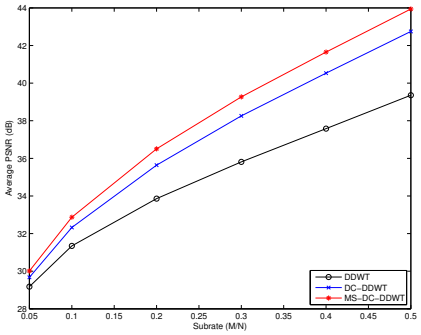


Fig. 5. Reconstruction quality for “Baby” as a function of subrate.

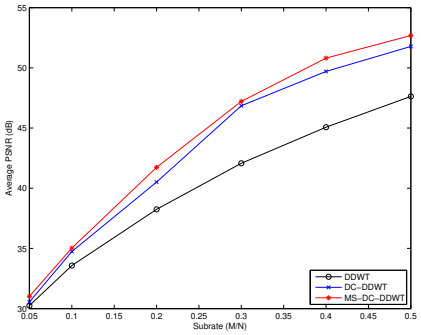


Fig. 6. Reconstruction quality for “Plastic” as a function of subrate.

Bibliography

Aaron, A., Girod, B. (2002). Compression with side information using turbo codes. *In Proc. Data Compression Conference*, pages 252–261, Snowbird, UT, USA.

Aaron, A., Rane, S., Girod, B. (2004a). Wyner-Ziv video coding with hash-based motion compensation at the receiver. *In Proc. Int. Conf. on Image Processing*, vol. 5, pages 3097–3100, Singapore.

Aaron, A., Rane, S., Setton, E., Girod, B. (2004b). Transform-domain Wyner-Ziv codec for video. *In Proc. SPIE Visual Commun. and Image Processing*, pages 520–528, San Jose, CA, USA.

Aaron, A., Setton, E., Girod, B. (2003). Towards practical Wyner-Ziv coding of video. *In Proc. Int. Conf. on Image Processing*, vol. 3, pages 869–872, Barcelona, Spain.

Aaron, A., Zhang, R., Girod, B. (2002). Wyner-Ziv coding of motion video. *In Proc. Asilomar Conference on Signals, Systems and Computers*, vol. 1, pages 240–244.

Adikari, A. B. B., Fernando, W. A. C., Weerakkody, W. A. R. J., Arachchi, H. K. (2006). A sequential motion compensation refinement technique for distributed video coding of Wyner-Ziv frames. *In Proc. Int. Conf. on Image Processing*, pages 597–600, Atlanta, GA, USA.

Alparone, L., Barni, M., Bartolini, F., Cappellini, V. (1996). Adaptively weighted vector-median filters for motion fields smoothing. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 4, pages 2267–2270, Atlanta, GA, USA.

Alvarez, L.and Weickert, J., Sanchez, J. (2000). Reliable estimation of dense optical flow fields with large displacements,. *International Journal of Computer Vision*, 39:41–56.

Areia, J., Ascenso, J., Brites, C., Pereira, F. (2007). Wyner-Ziv stereo video coding using a side information fusion approach. *In Int. Workshop on Multimedia Sig. Proc.*, pages 453–456, Chania, Greece.

Artigas, X., Angeli, E., Torres, L. (2006). Side information generation for multiview distributed video coding using a fusion approach. *In Norwegian Signal Proc. Symp. and Workshop*, pages 250–253, Reykjavik, Iceland.

Artigas, X., Ascenso, J., Dalai, M., Klomp, S., Kubasov, D., Ouaret, M. (2007a). The DISCOVER codec: Architecture, techniques and evaluation. *In Picture Coding Symposium (PCS)*, Lisbon, Portugal.

Artigas, X., Tarres, F., Torres, L. (2007b). Comparison of different side information generation methods for multiview distributed video coding. *In Int. Conf. on Sig. Process. and Multimedia Appl.*, Barcelona, Spain.

Artigas, X., Torres, L. (2005). Iterative generation of motion-compensated side information for distributed video coding. *In Proc. Int. Conf. on Image Processing*, vol. 1, pages 833–836, Genoa, Italy.

Ascenso, J., Brites, C., Pereira, F. (2005a). Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. *In EURASIP Conf. on Speech and Image Process., Multimedia Commun. and Serv.*, Smolenice, Slovak Republic.

Ascenso, J., Brites, C., Pereira, F. (2005b). Motion compensated refinement for low complexity pixel based distributed video coding. Como, Italy.

Ascenso, J., Brites, C., Pereira, F. (2006). Content adaptive Wyner-Ziv video coding driven by motion activity. *In Proc. Int. Conf. on Image Processing*, pages 605–608, Atlanta, GA, USA.

Ascenso, J., Pereira, F. (2007). Adaptive hash-based side information exploitation for efficient Wyner-Ziv video coding. *In Proc. Int. Conf. on Image Processing*, vol. 3, pages 29–32, San Antonio, TX.

Ascenso, J., Pereira, F. (2008). Advanced side information creation techniques and framework for wyner-ziv video coding. *J. on Visu. Commun. and Image Repr.*, 19:600–613.

Avudainayagam, A., Shea, J., Wu, D. (2008). Hyper-trellis decoding of pixel-domain Wyner-Ziv video coding. *IEEE Trans. on Circ. and Syst. for Video Technology*, 18:557 – 568.

Badem, M., Fernando, W., Martinez, J., Cuenca, P. (2009). An iterative side information refinement technique for transform domain distributed video coding. *In Int. Conf. on Multimedia and Expo.*, New York, NY, USA.

Bahl, L., Cocke, J., Jelinek, F., Raviv, J. (1974). Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. on Inform. Theory*, 20(2):284–287.

Bassi, F., Kieffer, M., Dikici, C. (2009). Multiterminal source coding of bernoulli-gaussian correlated sources. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, Taipei, Taiwan.

Bassi, F., Kieffer, M., Weidmann, C. (2008). Source coding with intermittent and degraded side information at the decoder. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, Las Vegas, NV, USA.

Berger, T. (1971). *Rate Distortion Theory: A Mathematical Basis for Data Compression.* Englewood Cliffs.

Berger, T., Longo, G. (1977). *Multiterminal Source Coding, Information Theory Approach to Communications.* New York.

Berrou, C., Glavieux, A. (1996). Near optimum error correcting coding and decoding: turbo-codes. *IEEE Trans. on Commun.*, 44:1261–1271.

Berrou, C., Glavieux, A., Thitimajshima, P. (1993). Near shannon limit error-correcting coding and decoding:turbo-codes. *In Proc. Int. Conf. on Communications*, vol. 2, pages 1064–1070, Geneva, Switzerland.

Bjontegaard, G. (2001). Calculation of average PSNR differences between RD curves. Rapport technique, 13th VCEG-M33 Meeting, Austin, TX, USA.

Borchert, S., Westerlaken, R., Gunnewiek, R. K., Lagendijk, I. (2007a). On extrapolating side information in distributed video coding. *In Picture Coding Symposium (PCS)*, Lisboa, Portugal.

Borchert, S., Westerlaken, R., Klein Gunnewiek, R., Lagendijk, R. (2007b). Improving motion compensated extrapolation for distributed video coding. *In Proc. Conf. of the Advanced School for Computing and Imaging.*, Heijen, the Netherlands.

Brites, C., Ascenso, Jo a., Quintas Pedro, J., Pereira, F. (2008). Evaluating a feedback channel based transform domain wyner-ziv video codec. *EURASIP J. on Sign. Proc.: Image Commun.*, 23(4):269–297.

Brites, C., Ascenso, J., Pereira, F. (2006a). Feedback channel in pixel domain wyner-ziv video coding : Myths and realities. *In Proc. Eur. Sig. and Image Proc. Conference*, Florence, Italy.

Brites, C., Ascenso, J., Pereira, F. (2006b). Improving transform domain Wyner-Ziv video coding performance. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 2, pages 525–528, Toulouse, France.

Brites, C., Ascenso, J., Pereira, F. (2006c). Modeling correlation noise statistics at decoder for pixel based Wyner-Ziv video coding. *In Picture Coding Symposium (PCS)*, Beijing, China.

Brites, C., Ascenso, J., Pereira, F. (2006d). Studying temporal correlation noise modeling for pixel based Wyner-Ziv video coding. *In Proc. Int. Conf. on Image Processing*, pages 273–276, Atlanta, GA, USA.

Brites, C., Pereira, F. (2007). Encoder rate control for transform domain wyner-ziv video coding. *In Proc. Int. Conf. on Image Processing*, San Antonio, TX, USA.

Brites, C., Pereira, F. (2008). Correlation noise modeling for efficient pixel and transform domain Wyner–Ziv video coding. *IEEE Trans. on Circ. and Syst. for Video Technology*, 18(9):1177–1190.

Cafforio, C., Rocca, F. (1983). The differential method for motion estimation. *In* Huang, T. S., éditeur : *Image Sequence Processing and Dynamic Scene Analysis*, pages 104–124.

Chang, P., Leou, J., Hsieh, H. (2001). A genetic algorithm approach to image sequence interpolation. *EURASIP J. on Sign. Proc.: Image Commun.*, 16(6):507–520.

Chen, S., Williams, L. (1993). View interpolation for image synthesis. *In Proc. Int. Conf. on Computer graphics and interactive techniques*, pages 279–287, Anaheim, CA, USA.

Chen, T. (2002). Adaptive temporal interpolation using bidirectional motion estimation and compensation. *In Proc. Int. Conf. on Image Processing*, Rochester, NY, USA.

Cohen, A., Daubechies, I., Feauveau, J.-C. (1992). Biorthogonal bases of compactly supported wavelets. *Comm. Pure Applied Math.*, 45(5):485–560.

Combettes, P. (2003). A block iterative surrogate constraint splitting method for quadratic signal recovery. *IEEE Trans. on Signal Proc.*, 51(7):1771–1782.

Cote, G., Erol, B., Gallant, M., Kossentini, F. (1998). H.263+: video coding at low bit rates. *IEEE Trans. on Circ. and Syst. for Video Technology*, 8:849 – 866.

Cover, T. (1975). A proof of the data compression theorem of slepian and wolf for ergodic sources. *IEEE Trans. on Inform. Theory*, 21:226 – 228.

Cover, T. M., Thomas, J. A. (2006). *Elements of Information Theory, Second Edition.* Hardcover.

Dalai, M., Leonardi, R., Pereira, F. (2006). Improving turbo codec integration in pixel-domain distributed video coding. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, Toulouse, France.

Daribo, I. (2009). *Codage et rendu de séquence vidéo 3D; et applications à la télévision tridimensionnelle (TV 3D) et à la télévision à base de rendu de vidéos.* Thèse de doctorat, TELECOM ParisTech, Paris, France.

Deligiannis, N., Munteanu, A., Clerckx, T., Cornelis, J., Schelkens, P. . (2009). On the side-information dependency of the temporal correlation in Wyner-Ziv video coding. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, Taipei, Taiwan.

Dinh, T. N., Lee, G., Chang, J.-Y., Cho, H.-J. (2007). A novel motion compensated frame interpolation method for improving side information in distributed video coding. *In Proc. Int. Symp. Information Theory*, pages 179–183, Joenju, South Korea.

DISCOVER-website (2005). www.discoverdvc.org.

Dufaux, F., Konrad, J. (2000). Efficient, robust, and fast global motion estimation for video coding. *IEEE Trans. on Image Proc.*, 9:497–501.

Esmaili, G., Cosman, P. (2009). Correlation noise classification based on matching success for transform domain Wyner-Ziv video coding. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, Taipei, Taiwan.

Ferre, P., Agrafiotis, D., Bull, D. (2007). Fusion methods for side information generation in multi-view distributed video coding systems. *In Proc. Int. Conf. on Image Processing*, San Antonio, TX, USA.

Fossorier, M., Lin, S. (1995). Soft-decision decoding of linear block codes based on ordered statistics. *IEEE Trans. on Inform. Theory*, 41:1379 – 1396.

Fowler, J. E. (2005). An implementation of PRISM using qccpack. Rapport technique, MSSU-COE-ERC-05-01, Mississippi State ERC, Mississippi State University.

Fraysse, A., Pesquet-Popescu, B., Pesquet, J. (2009). On the uniform quantization of a class of sparse source. *IEEE Trans. on Inform. Theory*, 55:3243–3263.

Gallager, R. (1963). Low density parity check codes. *MA: MIT Press*, 0(0). Cambridge.

Garcia-Frias, J., Zhao, Y. (2001). Compression of correlated binary sources using turbo codes. *IEEE Communication Letters*, 5:417 – 419.

Girod, B. (1993). What's wrong with mean-squared error? *Digital Images and Human Vision*, pages 207–220.

Girod, B., Aaron, A., Rane, S., Rebollo-Monedero, D. (2005). Distributed video coding. *Proc. IEEE*, 93(1):71–83.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Professional, 1 édition.

Gray, R. (1990). *Source coding theory.* Kluwer Academic Publisher.

Guillemot, C., Pereira, F., Torres, L., Ebrahimi, T., Leonardi, R., Ostermann, J. (2007). Distributed monoview and multiview video coding: Basics, problems and recent advances. *IEEE Signal Processing Magazine*, pages 67–76. Spec. Iss. on Sig. Process. for Multiterminal Commun. Syst.

Guo, X., Lu, Y., Wu, F., Gao, W., Li, S. (2006a). Distributed multi-view video coding. *In Proc. SPIE Visual Commun. and Image Processing*, vol. 6077, pages 15–19, San Jose, California, USA.

Guo, X., Lu, Y., Wu, F., Gao, W., Li, S. (2006b). Free viewpoint switching in multi-view video streaming using wyner-ziv video coding. *In Proc. SPIE Visual Commun. and Image Processing*, vol. 6077, pages 1–8, San Jose, California, USA.

Halloush, R., Radha, H. (2010). Practical distributed video coding based on source rate estimation. *In Proc. Conf. on Information Sciences and Systems*, Princeton, NJ, USA.

Huang, X., Forchhammer, S. (2008). Improved side information generation for distributed video coding. *In Int. Workshop on Multimedia Sig. Proc.*, Cairns, Queensland, Australia.

Huang, X., Forchhammer, S. (2009). Improved virtual channel noise model for transform domain Wyner-Ziv video coding. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*

Ingo Feldmann, I., Kauff, P., Mueller, K., Mueller, M., Smolic, A., Tanger, R., Wiegand, T., Zilly, F. (2008). HHI test material for 3D video. MPEG2008/M15413. Airchamps.

ISO/IEC MPEG & ITU-T VCEG (2007). Joint multiview video model (JMVM).

JPEG-2000 (2000). ISO/IEC FCD 15444-1: JPEG 2000 final comitee draft version 1.0.

Klomp, S., Vatis, Y., Ostermann, J. (2006). Side information interpolation with sub-pel motion compensation for wyner-ziv decoder. *In Int. Conf. on Sig. Process. and Multimedia Appl.*, Setubal, Portugal.

Kubasov, D. (2008). *Codage de sources distribuées : nouveaux outils et application à la compression vidéo.* Thèse de doctorat, Université de Rennes 1, IRISA, Rennes, France.

Kubasov, D., Guillemot, C. (2006). Mesh-based motion compensated interpolation for side information extraction in distributed video coding. *In Proc. Int. Conf. on Image Processing.*

Kubasov, D., Lajnef, K., Guillemot, C. (2007a). A hybrid encoder/decoder rate control for wyner-ziv video coding with a feedback channel. *In Int. Workshop on Multimedia Sig. Proc.*, Chania, Crete, Greece.

Kubasov, D., Nayak, J., Guillemot, C. (2007b). Optimal reconstruction in Wyner-Ziv video coding with multiple side information. *In Int. Workshop on Multimedia Sig. Proc.*, Chania, Crete, Greece.

Kullback, S., Leibler, R. (1951). On information and sufficiency. *Ann. of Mathematical Statiscics*, 22:79–86.

Lee, S., Kwon, O., Park, R. (2003). Weighted-adaptive motion-compensated frame rate up-conversion. *IEEE Trans. Consumer Electron*, 49:485–492.

Liveris, A.D. ; Zixiang Xiong ; Georghiades, C. . (2002). Compression of binary sources with side information at the decoder using ldpc codes. *IEEE Communication Letters*, 6:440 – 442.

Macchiavello, B., De Queiroz, R. L. (2007). Motion-based side-information generation for a scalable wyner-ziv video coder. *In Proc. Int. Conf. on Image Processing*, San Antonio, Texas, USA.

Mackay, D. J. C., Neal, R. M. (1997). Near shannon limit performance of low density parity check codes. *IEE Electronics Letters*, 33(6):457–458.

Macwilliams, F., Sloane, N. (1977). *The theory of Error Correcting Codes.* Elsevier.

Majumdar, A., Puri, R., Ishwar, P., Ramchandran, K. (2005). Complexity/performance trade-offs for robust distributed video coding. *In Proc. Int. Conf. on Image Processing*, vol. 2, pages 678–681, Genoa, Italy.

Mallat, S. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Anal. and Match. Int.*, 11(7):674–693.

Martinian, E., Behrens, A., Xin, J., Vetro, A. (2006). View synthesis for multiview video compression. *In Picture Coding Symposium (PCS)*, Beijing, China.

Martins, R., Brites, C., Ascenso, J., Pereira, F. (2009). Refining side information for improved transform domain wyner-ziv video coding. *IEEE Trans. on Circ. and Syst. for Video Technology*, 19:1327–1341.

Maugey, T., Miled, W., Cagnazzo, M., Pesquet-Popescu, B. (2009). Fusion schemes for multiview distributed video coding. *In Proc. Eur. Sig. and Image Proc. Conference*, Glasgow, Scotland.

Maugey, T., Pesquet-Popescu, B. (2008). Side information estimation and new symmetric schemes for multi-view distributed video coding. *J. on Visu. Commun. and Image Repr.*, 19(8):589–599. Special issue: Resource-Aware Adaptive Video Streaming.

Miled, W., Pesquet, J.-C., Parent, M. (2006). Disparity map estimation using a total variation bound. *In Canadian Conf. Comput. Robot Vis.*, pages 48–55, Quebec, Canada.

Miled, W., Pesquet, J.-C., Parent, M. (2009). A convex optimization approach for depth estimation under illumination variation. *IEEE Trans. on Image Proc.*, 18:813–830.

Morbee, M., Prades-Nebot, J., Pizurica, A., Philips, W. (2007). Rate allocation algorithm for pixel-domain distributed video coding without feedback channel. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 1, pages 521–524, Honolulu, Hawaii.

Müller, F. (1993). Distribution shape of two-dimensional dct coefficients of natural images. *Electronics Letters*, 29(22):1935–1936.

Nadarajah, S. (2005). A generalized normal distribution. *Journal of Applied Statistics*, 32:685–694.

Nagel, H., Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. on Pattern Anal. and Match. Int.*, 5:565–593.

Natario, L., Brites, C., Ascenso, J., Pereira, F. (2005). Extrapolating side information for low-delay pixel-domain distributed video coding. *In Int. Workshop on Visual Content Process. and Representation*, Sardinia, Italy.

Ouaret, M., Dufaux, F., Ebrahimi, T. (2006). Fusion-based multiview distributed video coding. *In Proc. ACM Int. Workshop on Video Surveillance and Sensor Networks*, Santa Barbara, California, USA.

Ouaret, M., Dufaux, F., Ebrahimi, T. (2007). Multiview Distributed Video Coding with Encoder Driven Fusion. *In Proc. Eur. Sig. and Image Proc. Conference*, Poznan, Poland.

Ouaret, M., Dufaux, F., Ebrahimi, T. (2009). Iterative multiview side information for enhanced reconstruction in distributed video coding. *EURASIP J. on Image and Video Proc.* special issue on Distributed Video Coding.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann.

Petrazzuoli, G., Cagnazzo, M., Pesquet-Popescu, B. (2010). High order motion interpolation for side information improvement in dvc. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, Dallas, Texas, USA.

Pradhan, S., Ramchandran, K. (1999). Distributed source coding using syndromes (DISCUS): design and construction. *In Proc. Data Compression Conference*, Snowbird, UT , USA.

Pradhan, S., Ramchandran, K. (2003). Distributed source coding using syndromes (discus): design and construction. *IEEE Trans. on Inform. Theory*, 49:626 – 643.

Puri, R., Majumdar, A., Ramchandran, K. (2007). PRISM: A video coding paradigm with motion estimation at the decoder. *IEEE Trans. on Image Proc.*, 16:2436–2448.

Puri, R., Ramchandran, K. (2002). PRISM: A new robust video coding architecture based on distributed compression principles. *In Proc. of the 40th Allerton Conference on Communication, Control and Computing*, Allerton, IL, USA.

Puri, R., Ramchandran, K. (2003). PRISM: A video coding architecture based on distributed compression principles. Rapport technique UCB/ERL M03/6, EECS Department, University of California, Berkeley.

Qing, L., He, X., Lv, R. (2007). Modeling non-stationary correlation noise statistics for Wyner-Ziv video coding. *In Int. Conf. on Wavelet Analysis and Pattern Recogn.*, Beijing, China.

Richardson, T. J., Shokrollahi, M. A., Urbanke, R. L. (2001). Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. on Inform. Theory*, 47(2):619–637.

Rudin, L., Osher, S., Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268.

Ryan, W. E. (1997). A turbo code tutorial. Rapport technique, New Mexico state University.

Scharstein, D., Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7âĂŞ–42.

Sheng, T., Hua, G., Guo, H, Z. J., Chen, C. (2008). Rate allocation for transform domain wyner-ziv video coding without feedback. *In Proc. ACM Int. Conf. on Multimedia*, Vancouver, British Colombia, Canada.

Sheng, T., Zhu, X., Hua, G., Guo, H, Z. J., Chen, C. (2010). Feedback-free rate-allocation scheme for transform domain Wyner-Ziv video coding. *J. Multimedia Systems*, 16:127–137.

Shum, H., Kang, S. (2000). A review of image-based rendering techniques. *In Proc. SPIE Visual Commun. and Image Processing*, vol. 4067 *2-13*, Perth , Australia.

Slepian, D., Wolf, J. K. (1973). Noiseless coding of correlated information sources. *IEEE Trans. on Inform. Theory*, 19(4):471–480.

Slowack, J., Mys, S., Skorupa, J., Lambert, P., Van de Walle, R., Grecos, C. . (2009). Accounting for quantization noise in online correlation noise estimation for distributed video coding. *In Picture Coding Symposium (PCS)*, Chicagi, IL, USA.

Stankovic, L., Stankovic, V., Wang, S., Cheng, S. (2010). Distributed video coding with particle filtering for correlation tracking. *In Proc. Eur. Sig. and Image Proc. Conference*, Aalborg, Denmark.

Stankovic, V., Liveris, A., Xiong, Z., Georghiades, C. (2006). On code design for the slepian-wolf problem and lossless multiterminal networks. *IEEE Trans. on Inform. Theory*, 52:1495 – 1507.

Tagliasacchi, M., Trapanese, A., Tubaro, S. (2006a). Exploiting spatial redundancy in pixel domain wyner-ziv video coding. *In IEEE Int. Conf. on Image Processing*, Atlanta, GA, USA.

Tagliasacchi, M., Tubaro, S. (2007). Hash-based motion modeling in wyner-ziv video coding. *In IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Honolulu, HawaI.

Tagliasacchi, M., Tubaro, S., Sarti, A. (2006b). On the modeling of motion in wyner-ziv video coding. *In Proc. Int. Conf. on Image Processing*, Atlanta, USA.

Tang, C., Au, O. (1998). Comparison between block-based and pixel-based temporal interpolation for video coding. *In Proc. Int. Symp. on Circ. and Syst.*, Monterey, CA , USA.

Taubman, D. (2000). High performance scalable image compression with ebcot. *IEEE Trans. on Image Proc.*, 9:1158–1170.

Toto-Zarasoa, V., Roumy, A., Guillemot, C. (2010). Non-uniform source modeling for distributed video coding. *In Proc. Eur. Sig. and Image Proc. Conference*, Aalborg, Denmark.

Varodayan, D., Aaron, A., Girod, B. (2005). Rate-adaptive distributed source coding using low-density parity-check codes. *In Proc. Asilomar Conference on Signals, Systems and Computers*, Monterey, Californa, USA.

Varodayan, D., Chen, D., Flierl, M., Girod, B. (2008). Wyner-ziv coding of video with unsupervised motion vector learning. *EURASIP J. on Sign. Proc.: Image Commun.*, 23:369–378.

Wang, P., Liu, X. (2009). A parallel algorithm for side information generation in distributed video coding. *In Int. Symp. on Indust. Electro.*, Seoul, South Korea.

Wang, Z., Bovik, A. (2009). Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Trans. on Signal Proc.*, 26:98–117.

Weerakkody, W., Fernando, W., Martinez, J., Cuenca, P., Quiles, F. (2007). An iterative refinement technique for side information generation in DVC. *In Int. Conf. on Multimedia and Expo.*, Beijing, China.

Wiegand, T., Sullivan, G., Bjontegaard, G., Luthra, A. (2003). Overview of the H.264/AVC video coding standard. *IEEE Trans. on Circ. and Syst. for Video Technology*, 13(7):560–576.

Wyner, A. (1974). Recent results in the shannon theory. *IEEE Trans. on Inform. Theory*, 20:2–10.

Wyner, A. (1975). On source coding with side information at the decoder. *IEEE Trans. on Inform. Theory*, 21:294–300.

Wyner, A., Ziv, J. (1976). The rate-distortion function for source coding with side information at the receiver. *IEEE Trans. on Inform. Theory*, 22:1–11.

Xu, Q., Xiong, Z. (2006). Layered Wyner-Ziv video coding. *IEEE Trans. on Image Proc.*, 15(12):3791–3803.

Yaacoub, C., Farah, J., Pesquet-Popescu, B. (2008). Feedback channel supression in distributed video coding with adaptative rate allocation and quantization for multi-sensor applications. *EURASIP J. on Wireless Commun. and Networking*, 2008:1–13.

Yaacoub, C., Farah, J., Pesquet-Popescu, B. (2009a). A genetic algorithm for side information enhancement in distributed video coding. *In Proc. Int. Conf. on Image Processing*, Cairo, Egypt.

Yaacoub, C., Farah, J., Pesquet-Popescu, B. (2009b). A genetic frame fusion algorithm for side information enhancement in wyner-ziv video coding. *In Proc. Eur. Sig. and Image Proc. Conference*, Glasgow, Scotland.

Yaacoub, C., Farah, J., Pesquet-Popescu, B. (2009c). Improving hash-based wyner-ziv video coding using genetic algorithms. *In Int. Mobile Multimedia Commun. Conf. (MOBIMEDIA)*, London, UK.

Yang, Y., Stankovic, V., Xiong, Z., Zhao, W. (2008). On multiterminal source code design. *IEEE Trans. on Inform. Theory*, 54:2278 – 2302.

Ye, S., Ouaret, M., Dufaux, F., Ebrahimi, T. (2008). Improved side information generation with iterative decoding and frame interpolation for distributed video coding. *In Proc. Int. Conf. on Image Processing*, San Diego, USA.

Yeung, R., Zhang, Z. (1999). Distributed source coding for satellite communications. *"tit"*, 45:1111 – 1120.

Zhai, J., Yu, K., Li, J., Li, S. (2005). A low complexity motion compensated frame interpolation method. *In Proc. Int. Symp. on Circ. and Syst.*, Kobe, Japan.