

CHAPITRE 1

RÉSULTATS ET ANALYSE

~~Les~~ Au chapitre précédent, nous avons présenté les concepts théoriques des nouvelles approches sélectives proposées, basées sur le MCB et le SDMCB ~~ont été démontrées théoriquement au chapitre précédent. Maintenant, nous effectuons~~

[SC]: On utilise des acronymes français ou anglais pour cela ?

~~. Dans ce chapitre, nous faisons état~~ des essais pratiques ~~afin d'attester des réalisés pour valider les~~ gains réalisables à l'aide de ces approches dans des conditions ~~similaires à celles rencontrées par des applications~~ de vidéo mobile. ~~Dans ce chapitre, nous~~ Nous démontrons, non seulement l'efficacité des approches sélectives proposées, mais aussi ~~sa leur~~ supériorité par rapport à l'approche de dissimulation utilisée par le décodeur inclus avec le logiciel de référence H.264/AVC JM.

~~À la , nous présentons~~ Nous débutons par présenter nos hypothèses de validation à la section 1.1. Nous expliquons ensuite, à ~~titre d'exemple, des trames issues de la 726^e sous-séquence de notre jeu de tests. Les~~ la section 1.2, les composantes de notre banc d'essai ~~, utilisé utilisées~~ pour créer ~~ce notre~~ jeu de tests, ~~sont expliquées à la section 1.2. L'image erronée permet la visualisation du résultat obtenu par le.~~ Dans le but de valider nos hypothèse sur le décodeur, les taux de succès observés lors du décodage d'une ~~tranche corrompue. Les probabilités de réussir un tel décodage~~ séquence erronée sont présentées à la section 1.3 ~~. Tandis qu'~~

[SC]: je ne parle pas de probabilité car c'est un événement mesuré, pas théorique

~~. Un succès sera considéré lorsque le décodeur retournera une image sans planter~~

[SC]: trouver mieux que planter

. Finalement, à la section 1.4, nous ~~mesurons~~ présentons les résultats obtenus de l'application de nos algorithmes de détection et dissimulation d'erreurs sur ~~cette image~~ les images décodées.

1.1 Hypothèses de validation

Débutons par énumérer les hypothèses faites dans le cadre de nos expérimentations. À la figure 1.1, nous présentons, à titre d'exemple, des trames issues de la 726^e sous-séquence de notre jeu de tests (expliqué plus en détail à la section suivante). L'image erronée (b) correspond au résultat obtenu par le décodage d'une tranche corrompue.



Figure 1.1 Les trames résultantes de la sous-séquence #726 du jeu de tests (Séquence : Foreman, QP=16 BER=0.0008, FMO=Dispersé)

[SC]: pour la figure 1, il me semble que tu ne peux pas mettre les descriptions dans les figures et il fait les mettre dans la description globale de la figure

La trame qui précède l'erreur (a) ne contient jamais d'erreurs, une ~~supposition~~hypothèse posée aussi par (?). Nous ~~avons~~aurons recours à cette trame comme source de contenu pour la dissimulation d'erreurs de la trame (b). La trame (c) est celle reconstruite à l'aide du calquage de la tranche de la trame précédente (a). Cette trame représente la solution d'un décodeur moderne, ce à quoi nous allons comparer nos dissimulations. La figure (d) est la trame de référence, qui n'a pas subi l'encodage. C'est la trame que nous utilisons pour mesurer le PSNR.

[SC]: mettre toutes tes hypothèses ici

~

1.2 Description du banc d'essai

Afin ~~d'étudier nos hypothèses sur la détection de la vraisemblance de régions d'images~~valider notre notre approche de détection et de dissimulation vidéo, nous avons construit un banc d'essai constitué d'un grand nombre de

[SC]: il me semble que la phrase d'avant n'avait pas de sens. il me semble que tu n'étudies pas tes hypothèses, enfin ce n'est pas le but premier du banc d'essai. De plus tu n'étudies pas tes hypothèses sur la détection de la vraisemblance de régions d'images. C'est quoi une détection de la vraisemblance ? On fait une détection ou on évalue la vraisemblance mais il me semble qu'on ne fait pas une détection de la vraisemblance

séquences vidéos encodées à divers débits et soumises à des taux d'erreurs ~~;~~ similaires à ceux proposés dans l'ouvrage (?). Ces ~~taux représentent les conditions présentes lors du~~découlement des conditions difficiles de transport de séquences vidéo H.264 vers des appareils évoluant sur des réseau mobiles.

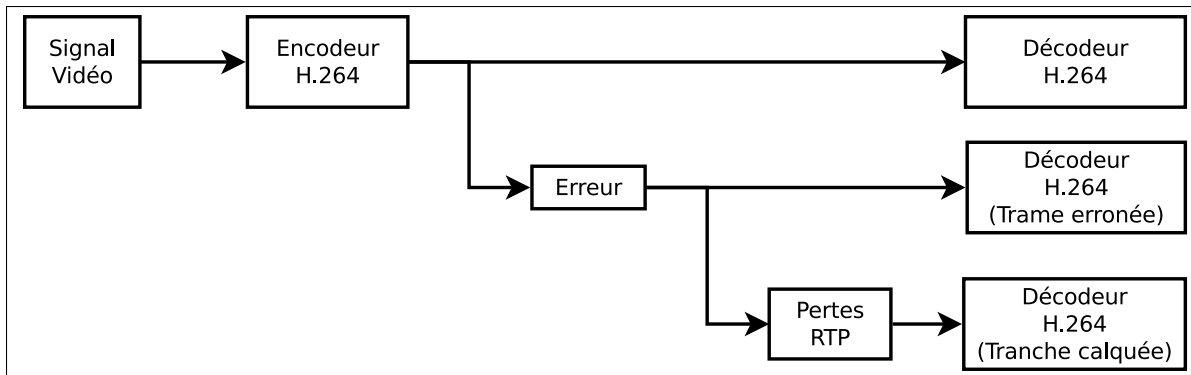


Figure 1.2 Diagramme des étapes du banc d'essai

[SC]: les boîtes des 3 décodeurs devrait être toutes pareilles et chacune devrait avoir une sortie avec un nom différent : signal vidéo sans erreur, signal vidéo erroné, signal vidéo calqué ou quelque chose du genre

Nous présentons, à la figure 1.2, les étapes de notre banc d'essai. ~~En résumé,~~ Nous y observons qu'un signal vidéo est encodé, transmis, puis décodé trois fois. La première opération de décodage décode les données sans que celles-ci soient exposées à l'erreur. Cette étape est cruciale afin de déterminer la dégradation visuelle engendrée par l'encodage du signal. ~~Par la suite~~

[SC]: on mesure vraiment l'effet de l'encodage ou bien ça sert plutôt de référence pour mesurer les erreurs de transport

. Dans la seconde opération, la séquence encodée est sujette à un motif d'erreur binaire (*Bit Error Pattern*) gaussien. Les données corrompues sont envoyées au décodeur, ~~s.~~ S'il réussit à les décode, celles-ci sont conservées, sinon la sous-séquence est rejetée. En dernier lieu, la sous-séquence corrompue est analysée par un module de détection de pertes RTP (*RTP loss*). Celui-ci identifie et retire les paquets RTP corrompus. Les données résultantes de l'analyse de pertes RTP sont décodées. Lorsque le décodeur s'aperçoit d'un paquet manquant, il ~~procède~~ ~~à utiliser~~ utilise une approche de dissimulation d'erreurs, ~~soit~~ en remplaçant, par exemple, la

tranche manquante par son équivalent dans la trame précédente¹ (*slice copy*, traduit dans cet ouvrage par la locution tranche calquée).

[SC]: revoir cette phrase

Les séquences sont encodées et décodées à l'aide de la version 16.2 du logiciel de référence H.264/AVC JM (?). Ce dernier, n'étant pas un produit commercial, a pour objectif principal la démonstration de nouvelles fonctionnalités visant la norme H.264. Notre banc d'essai est constitué de sous-ensembles de quatre trames consécutives commençant à un emplacement aléatoire dans diverses séquences vidéos. Ces sous-ensembles débutent par une trame *intra* et trois trames *inter* (IPPP). Nous retenons cinq sous-ensembles pour chaque séquence de résolution QCIF contenue dans l'ensemble de séquences de référence de l'Université de l'Arizona (?).

[LT]: Définir QCIF dans la section transport de vidéo.

Dans le contexte d'applications mobiles, le débit est souvent fixe (typiquement entre 64 kb/s , et 128 kb/s ...pour les séquences QCIF). Cependant, pour simplifier nos expérimentations, nous imposons plutôt un pas de quantification (QP) fixe. Ceci élimine les variations de **PSNR** dû qualité, mesurées à l'aide du PSNR

[LT]: Définir PSNR

, dus aux changements de QP requis pour garder le débit fixe. Les indices de pas de quantifications utilisés sont : 16, 20, 24 et 28. Ces valeurs sont choisies, car elles représentent un intervalle de données plausibles pour une application mobile selon la bande passante disponible. De plus, nous n'avons pas retenu des QP trop élevés (>30), car la quantité de bits requis

1. Dans cet ouvrage, *slice copy* a été utilisé au détriment de *motion copy*, car aucune implémentation fonctionnelle de *motion copy* était disponible. Nous avons eue des problèmes lors de l'utilisation de *motion copy* avec les versions 15.1 à 16.2 du décodeur inclus dans le logiciel de référence H.264/AVC JM. Ces problèmes sont connus et confirmés par plusieurs forums internet et catalogués dans le logiciel de suivi de problèmes Mantis relié au logiciel de référence H.264/AVC JM (?). Quoique notre ouvrage est aussi compatible avec *motion copy*, c'est pour cette raison, que nous avons testé uniquement avec *slice copy*.

pour encoder une trame est tellement faible qu'il arrive souvent, que les taux d'erreurs utilisés ne parviennent pas à briser un seul bit de la trame.

[SC]: mettre une petite note que les QP varient entre 0 et 51

Les taux d'erreur binaire (BER) utilisés sont : 0.0004, 0.0008, 0.0016, et 0.0032. Cet intervalle de taux est considérablement large et agressif par rapport aux intervalles d'erreurs retrouvés dans la littérature de la norme H.264 (?). Nous avons ~~pris~~choisi un tel intervalle pour obtenir des résultats témoignant des gains possibles dans un grand nombre de conditions d'erreur de transport. Le taux d'erreur utilisé est appliqué au niveau des bits et non pas au niveau des paquets (?) ou des blocs (comme dans les ouvrages de dissimulation d'erreurs).

Seulement la troisième trame (une trame P) est exposée à l'erreur. En ce qui concerne l'ordonnement de macroblocs flexible (FMO), deux types sont employés : dispersé (*dispersed*) et entrelacé (*interlaced*). Tous deux sont composés de deux tranches. À la figure 1.3 et ~~1.1~~1.4, nous présentons deux exemples d'erreurs pour chacun des types d'ordonnement. Les sous-séquences sont encodées à 30 images par seconde, tandis que les autres paramètres sont propres au profil de base (*baseline*) défini dans la norme H.264. Le format de sortie est RTP et la résolution est QCIF, c'est-à-dire 176×144 pixels. Cependant, les en-têtes NAL et RTP n'ont pas été exposés à l'erreur.

[SC]: Il me semble que les figures sont inversées. La première est interleaved (entrelacé) et la seconde dispersed

Notre banc d'essai est constitué d'un jeu de tests de 2720 sous-ensembles de trames, Plus précisément, nous avons 1360 trames dispersées et 1360 trames entrelacées. Dans la prochaine section, ~~il sert ce~~ce banc d'essai servira à déterminer la résilience du décodeur inclus dans le logiciel de référence H.264/AVC JM. À la section 1.4, il ~~est~~sera utilisé pour valider nos approches sélectives de détection et de dissimulation d'erreur.

[SC]: Tu dois expliquer comment tu arrives à 1360. $1360 = N \text{ séquences} \times 4 \text{ taux erreur} \times 4 \text{ QP} \times \text{etc.}$

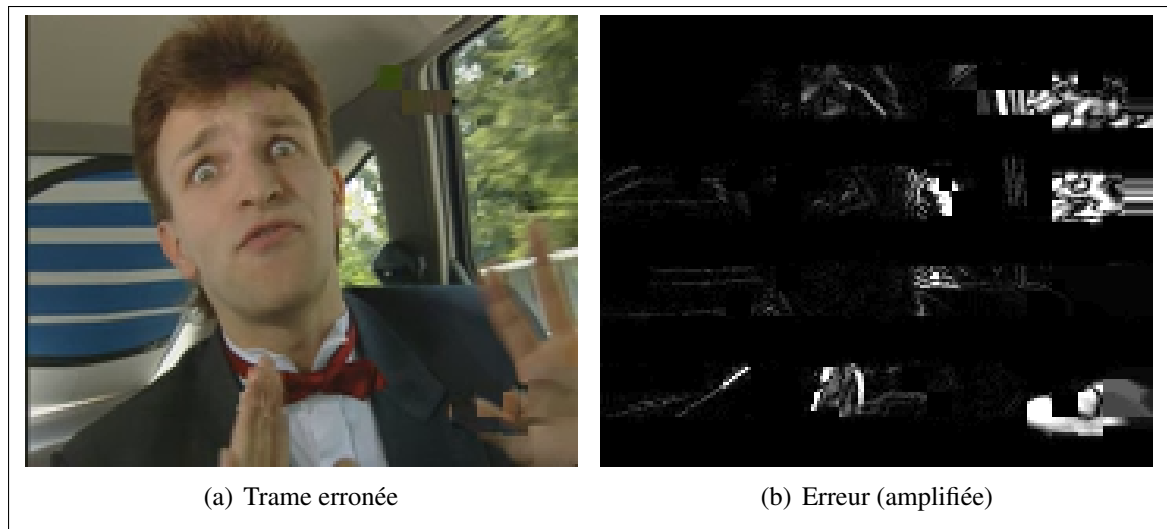


Figure 1.3 Exemple de l'erreur présente dans la sous-séquence #392 du jeu de tests (Séquence : Carphone, QP=16 BER=0.0032, FMO=Dispersé)

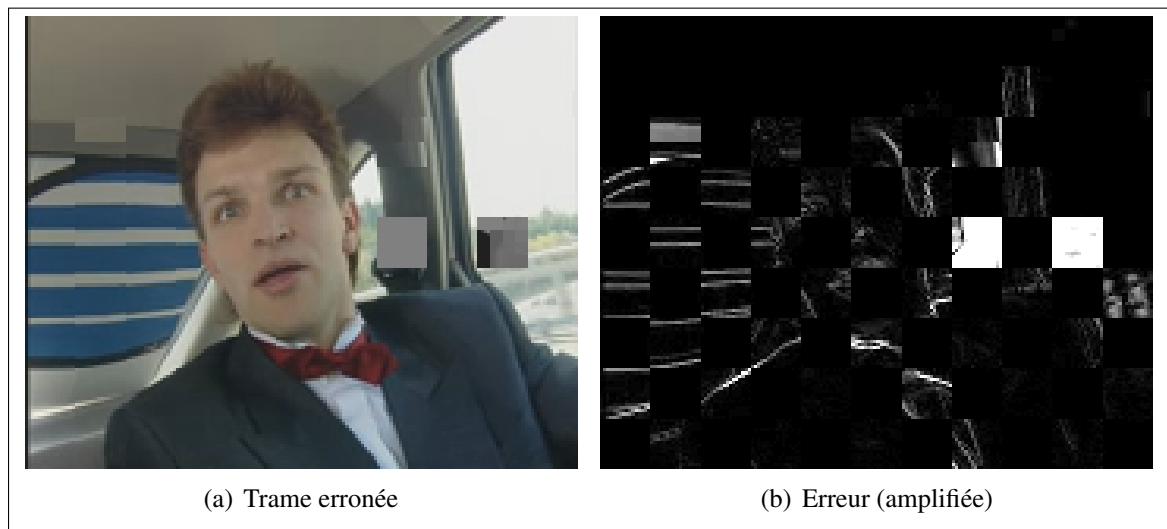


Figure 1.4 Exemple de l'erreur présente dans la sous-séquence #1752 du jeu de tests (Séquence : Carphone, QP=16 BER=0.0032, FMO=Entrelacé)

[SC]:!!!! Très important!!!! Ta méthodologie quant aux séquences utilisées pour entraîner ton algorithme (e.x. trouver le bon seuil) versus établir les performances sera scrutée à la loupe. Tu ne peux pas entraîner sur le même contenu sur lequel tu mesures les performances. Tu dois présenter sur quoi tu entraînes et sur quoi tu évalues la performance.

1.3 Analyse de la résilience aux erreurs du décodeur de référence H.264

Dans certaines conditions, le décodeur fourni avec le logiciel de référence H.264/AVC JM réussit à décoder des séquences binaires corrompues. Cette découverte, initiatrice de notre effort de recherche, fut réalisée très tôt dans nos expérimentations. Nous définissons un décodage réussi comme : l'opération de décoder une tranche corrompue sans causer de défaillances (*crash*) chez le décodeur, sans pour autant garantir l'intégrité ~~de~~ du contenu résultant. En d'autres mots, l'image issue du décodage d'une tranche corrompue va souvent contenir de la dégradation visuelle (voir figures 1.1(b), 1.3(a) et 1.4(a)). Le taux de décodages réussis varie, non seulement selon le taux d'erreur binaire appliqué à la séquence, mais aussi à l'égard du paramètre de quantification (QP) utilisé lors de l'encodage. Ceci s'explique par le fait qu'un QP plus faible ~~requiert~~ force l'encodeur à utiliser plus de bits pour encoder une trame, ~~;~~ et plus il y a de bits sujets à ~~l'erreur~~, erreur et plus la probabilité de corruption de la trame augmente.

[SC]: Ajuster les axes des y car je ne crois pas que les taux de réussites soient de 0.3 pourcent

~~L'histogramme~~ Les histogrammes de la figure 1.5 ~~décrit~~ présentent les taux de décodages réussis observés en fonction des paramètres de quantification et du taux d'erreurs ~~binaires sur les bits~~ présents dans notre jeu de tests. Ces taux sont intéressants, car ils confirment que le décodage des données corrompues peut avoir un taux de réussite allant de 20 % à 70 %, selon les conditions. On peut concevoir une stratégie de résilience à l'erreur, où une prochaine génération de décodeurs ~~serait~~ seraient en mesure d'effectuer une tentative de décodage d'une tranche corrompue, dans un environnement contrôlé (possiblement un second décodeur), afin de contrer l'impact d'une défaillance.

[SC]: Pas clair, clarifier l'idée. c'est quoi contrer l'impact de la défaillance

Lors du décodage d'une tranche corrompue, le décodeur inclus dans le logiciel de référence H.264/AVC JM fait la transition entre trois états distincts. Le résultat de ceux-ci est illustré à la figure 1.6. Le premier état est celui du décodage des bits de la tranche corrompue situés avant

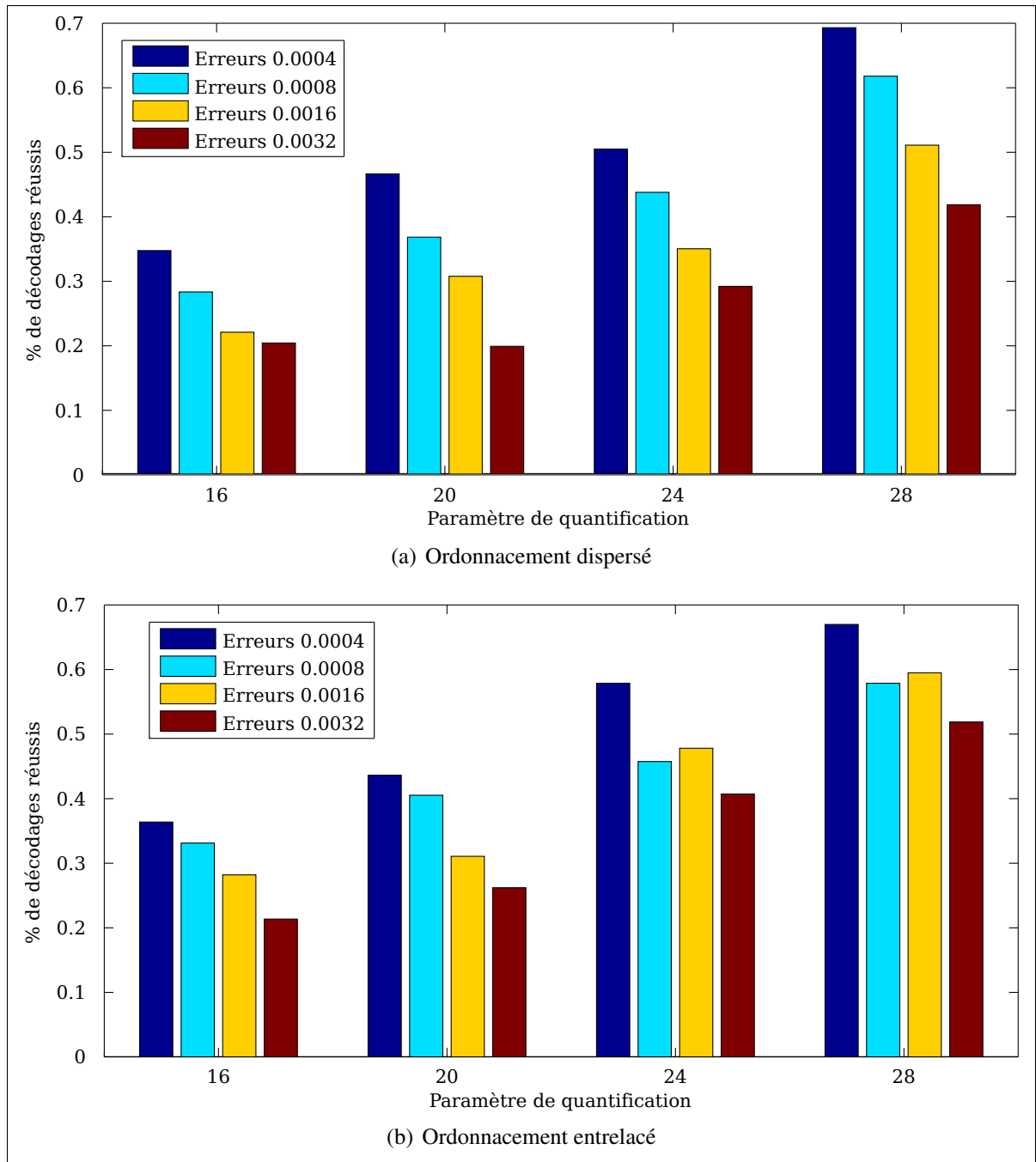


Figure 1.5 Histogrammes des pourcentages de décodages réussis en fonction du paramètre de quantification utilisé lors de l'encodage ainsi que du taux d'erreurs binaires sur les bits (BER).

l'erreur. Cet état est équivalent au décodage normal d'une trame. Il est illustré par la partie supérieure, tout en noir (signifiant l'absence de différence), du différentiel (b) de la figure 1.6.

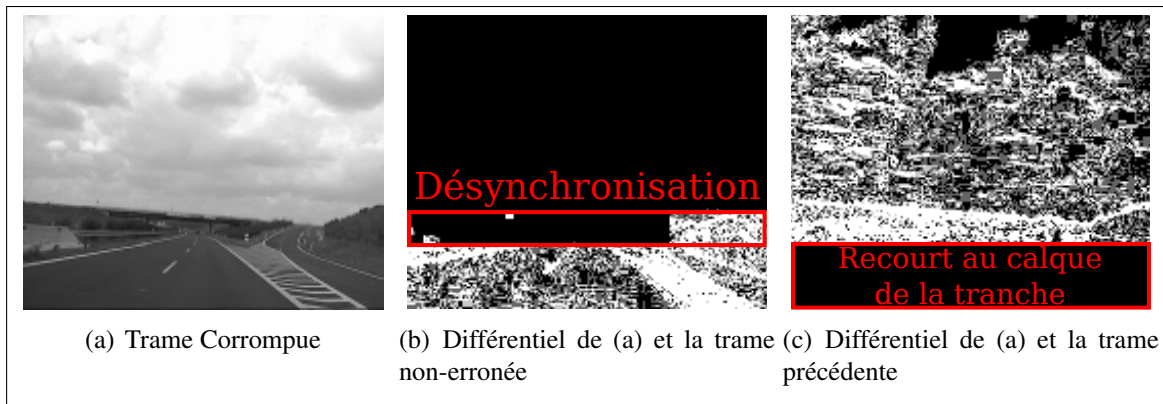


Figure 1.6 Exemple du comportement du décodeur. En (b), le différentiel de (a), par rapport à la trame ~~encodée, mais non-erronée~~codée/décodée sans erreur, est utilisé pour démontrer la désynchronisation. Le différentiel entre (a) et la trame précédente permet de démontrer, en (c), l'utilisation du calque de tranche.

Par la suite, le décodeur décode le premier bit corrompu. Ceci cause une désynchronisation des codes binaires de l'encodage entropique. Même si les autres bits de la séquence sont valides, la désynchronisation fait en sorte qu'ils sont associés à la mauvaise valeur dans la table de référence CAVLC. Le décodeur interprète ces mauvaises valeurs ~~ce qui entraîne~~ ; ce qui entraîne une dégradation visuelle dans l'image. Cette dégradation peut se manifester fortement comme c'est le cas dans l'image de la figure 1.1(b) ou être presque imperceptible, comme ~~eelle~~-c'est le cas dans la figure 1.6(a). Afin de mieux voir la dégradation visuelle de la figure 1.6(a), elle est encadrée à la figure 1.6(b). C'est dans ce second état que le décodeur est le plus vulnérable aux défaillances.

Le second état se termine lorsque le décodeur cesse d'insérer de la dégradation visuelle dans l'image et utilise le contenu provenant de la tranche précédente. Ce changement est illustré en 1.6(c). Ici, on suppose que le décodeur est complètement désynchronisé, il ne sait que faire de ce qu'il décode, il applique donc le contenu des blocs précédents, ce qui, grâce à la corrélation temporelle, est le contenu de remplacement correct, s'il y a absence de mouvement. De plus, notons que dans l'image 1.6(a), il y a très peu d'effet de blocs causés par la désynchronisation. Ceci est dû au filtre antiblocs, ~~lorsqu~~, Lorsqu'il est appliqué sur la trame, il va identifier et lisser les effets de blocs. Ceci explique aussi pourquoi, à la figure 1.3(a), l'erreur se répand à

la tranche non erronée. Dans ce cas, le filtre antiblocs lisse le bloc erroné, ce qui répand l'erreur dans les blocs avoisinants.

[SC]: Je ne suis pas trop certain de l'explication que le décodeur applique les blocs précédents. Est-ce vraiment ce qui se passe ? C'est pas que le décodeur a un buffer d'image décodé et qu'il l'écrase lors du décodage d'une nouvelle trame. S'il abandonne avant de terminer le décodage d'une trame, il reste du contenu résiduel appartenant à la trame précédente.

1.4 Analyse de l'approche sélective

Comme démontré dans la section précédente, l'image résultante ~~de la d'une~~ tranche corrompue peut souvent être décodée, ~~ce qui la rend disponible pour guider un algorithme de dissimulation d'erreur~~. Cependant, celle-ci possède, dû à la désynchronisation du décodeur lors du décodage, une ~~certaine~~ dégradation visuelle. ~~Un plus ou moins importante. Elle peut toutefois être utilisée pour guider un algorithme de dissimulation d'erreur grâce à un~~ nouveau type d'algorithme, ~~celui~~ permettant la détection de la dégradation visuelle ~~est requis~~ proposé dans ce mémoire.

L'algorithme de détection basé la mesure des effets de blocs compensés par le mouvement (MCB), présenté dans cet ouvrage, permet la détection d'un type de dégradation visuelle, celui causant des effets de blocs absents de la trame précédente. La prémisse de notre approche sélective est de déterminer entre le calque de la tranche (figure 1.1(c)) et la tranche corrompue (figure 1.1(b)), celui qui produirait la meilleure dissimulation (~~maximize PSNR~~ c.-à-d. produisant le meilleur PSNR par rapport à la trame de référence

[SC]: trame décodée sans erreur ou la trame originale ?

). Cette opération peut être accomplie à deux niveaux : celui de la tranche, à l'aide du SDMCB, ou des blocs, avec le MCB. Nous allons tout d'abord présenter la configuration de notre banc de tests pour mesurer l'approche sélective. Par la suite, la sous-section 1.4.1 présente les résultats obtenus avec le SDMCB et les résultats du MCB sont présentés à la sous-section 1.4.2.

[LT]: MCB, SDMCB et le PSNR sont présentés dans des sections précédentes.

[SC]: Les termes bancs d'essai utilisé à la section 1.1 et bancs de tests utilisés ici sont mélangeant pour le lecteur. Il faut clarifier chacun. Par exemple données d'essai devrait être utilisé à la section 1.1 dans le titre et banc d'essai pour la section actuelle. Le banc d'essai utilise les données d'essai. Le terme banc de tests n'existe pas dans le grand Dic

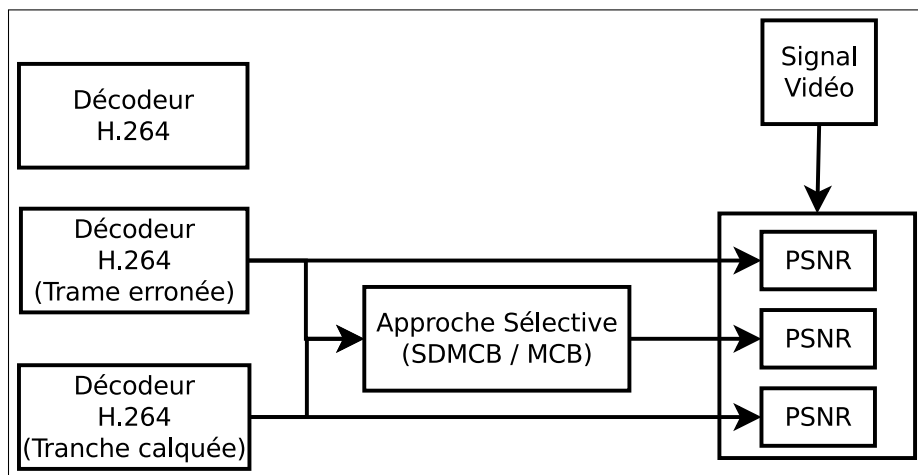


Figure 1.7 Partie 2 de la Figure 1.2. Configuration du banc d'essai pour mesurer l'approche sélective basée SDMCB et MCB.

[SC]: Enlever la référence à la figure 1.2. le mentionner dans le texte toutefois. Aussi voir mes commentaires sur la figure 1.2 et les appliquer ici pour le nom décodeur partout ex PSNRerr, PSNRTC, etc. Aussi, trouver des noms aux valeurs de PSNR de chaque signal (faire une flèche qui sort des boites PSNR)

Tout d'abord, à l'aide du PSNR comme discriminant, nous mesurons la distribution

[SC]: on calcule vraiment une distribution ?

des meilleures dissimulations, entre le calque de la tranche et la tranche corrompue. La figure 1.7 démontre la configuration élaborée pour obtenir ces mesures. On y **comprend** constate que le PSNR est mesuré par rapport au signal vidéo de référence et qu'il est mesuré pour ➤

l'image résultante du décodage de la tranche corrompue, le candidat de dissimulation basée sur le calque de la tranche et le résultat des approches sélectives basées sur le SDMCB et MCB.

Les résultats mesurés sur le banc de tests proposé sont présentés à la figure 1.8. On y ~~aperçoit~~ ~~qu'~~remarque une importante quantité de trames et de blocs erronés qui ~~démontrent~~ possèdent un PSNR supérieur aux calques de trames. ~~Ce qui~~ Cela s'explique par le fait ~~, que~~ que, dans ces cas, la dégradation visuelle engendrée par la désynchronisation du décodeur est moins importante que la variation temporelle par rapport à la trame précédente ~~exploitée par~~ ; ce qui pénalise le calque de tranche. En lien avec les trois états d'un décodeur ~~décodant~~ traitant une tranche corrompue, nous pouvons conclure que dans l'état 1, la tranche endommagée va produire un meilleur résultat que le calquage de la tranche. Dans l'état 2, la tranche endommagée va produire un résultat inférieur à la tranche calquée. Finalement, à l'état 3, les deux sont équivalents. La proportion de blocs compris dans les états 1 et 2, l'intensité de la dégradation produite par le décodeur ainsi que la variation temporelle par rapport à la trame précédente sont les facteurs qui déterminent laquelle des deux trames ~~sera~~ constituera la meilleure dissimulation.

[SC]: On veut vraiment dire la meilleure dissimulation ou plutôt la meilleure alternative ?

Plus précisément, les mesures obtenues de notre banc d'essai nous permettent de conclure que l'image résultante du décodage de la tranche corrompue offre un PSNR plus élevé dans 44 % des cas pour l'ordonnancement dispersé et 49 % pour l'entrelacé. Pour ces trames, on observe respectivement des gains moyens de 2.12 dB et 1.68 dB par rapport à leurs trames équivalentes ~~calquer. Tandis que~~ calquées. De manière similaire, les blocs issus de trames corrompues sont supérieurs dans 42 % des cas pour l'ordonnancement dispersé et 49 % pour l'entrelacé. Ces blocs produisent, pour les approches d'ordonnancement précédentes, des gains moyens de 1.44 dB et 1.02 dB.

[SC]: devrait-on avoir plus de graphiques en annexe comme montrer cela par débit ou taux d'arreur et voir la tendance ?

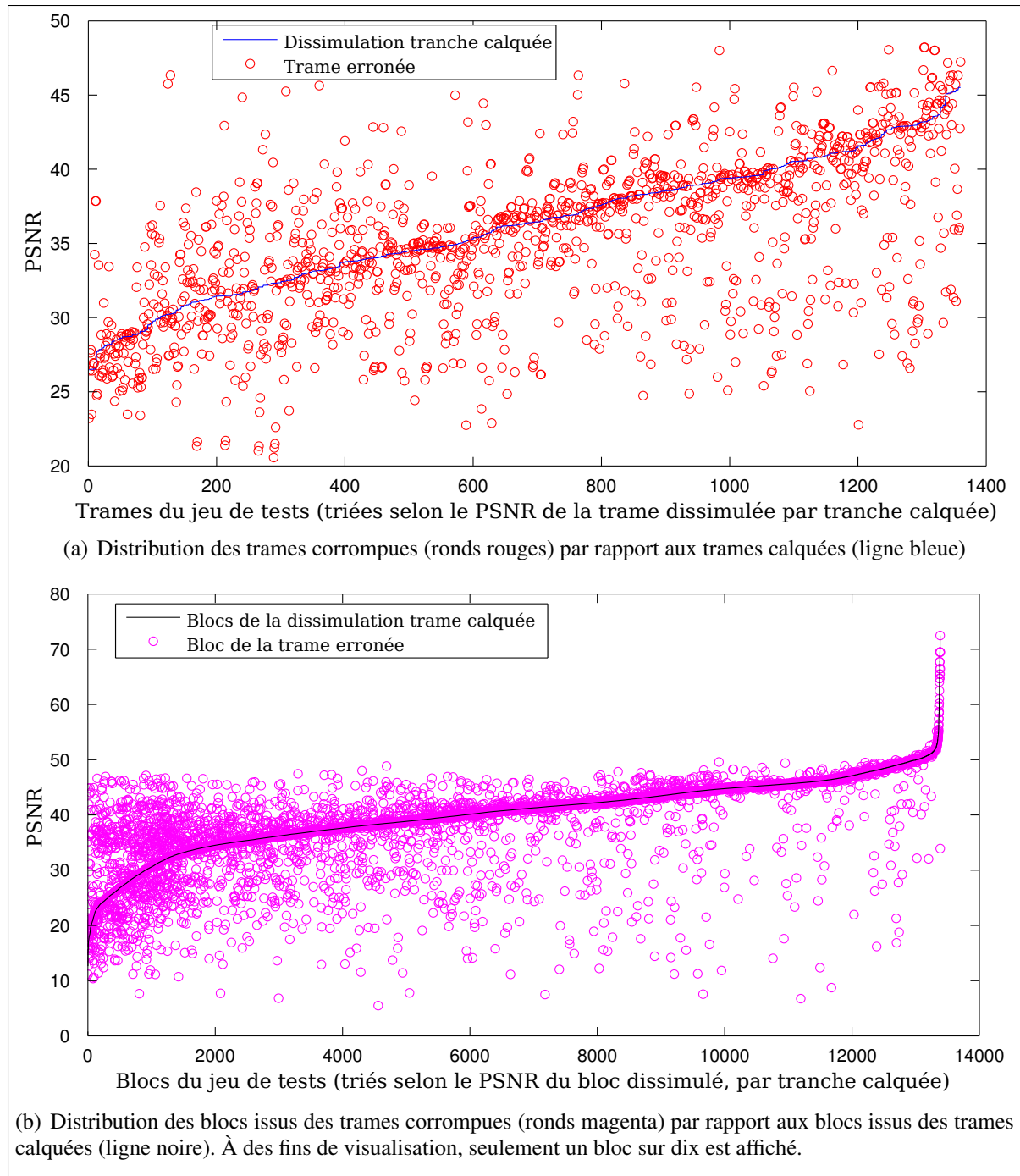


Figure 1.8 Visualisation de la distribution des valeurs du PSNR de la trame erronée et de la dissimulation par tranche calquée.

Nous avons démontré, non seulement, qu'il est possible de réussir à décoder des tranches erronées. Mais aussi qu'une quantité importante des images résultantes exhibent une fidélité vi-

suelle supérieure au calque de cette tranche (approche couramment employée par le décodeur fourni avec le logiciel de référence H.264/AVC JM).

1.4.1 Approche sélective SDMCB

Maintenant, toujours avec la même configuration du banc de test, mesurons l'aptitude du SDMCB à identifier le meilleur candidat entre la trame corrompue et la dissimulation par tranche calquée. Pour ce faire, nous allons utiliser une taille de bloc $B = 16$ (vu notre indépendance au train de bits, nous ne connaissons pas la taille exacte des blocs) et un seuil d'effet de bloc $T_b = 5000$.

[SC]: Les gens vont te questionner sur l'origine du 5000. Il faut expliquer. pour la taille des blocs, les MB 16x16 sont les plus souvent utilisés peu importe la norme. donner aussi la section dans laquelle la méthode SDMCB est présentée dans le mémoire. Quelques screenshot cas de trame calquée vs ton algo serait instructif

[SC]: Ronds ou cercles ?

Les résultats obtenus sont présentés à la figure 1.9. La figure 1.9(a) témoigne que l'approche sélective, dans le cadre d'un ordonnancement dispersé, fait le bon choix dans 81 % des cas. Pour ceux-ci, un gain moyen de 1.98 dB est mesuré, tandis que pour l'ensemble de la séquence le gain moyen est de 0.72 dB. Les choix de l'approche sélective offrent un PSNR supérieur aux trames dissimulées par calquage de tranche dans 31 % des cas. Pour la figure 1.9(b), la bonne trame est choisie dans 86 % des cas, résultant dans un gain moyen de 1.24 dB pour celles-ci. Sur l'ensemble des trames, le gain moyen est de 0.65 dB et un PSNR supérieur au calquage de trame est obtenu dans 43 % des cas.

1.4.2 Approche sélective MCB

Le niveau de granularité plus fin de l'approche sélective MCB est son attrait principal. En remplaçant uniquement les blocs endommagés, on maximise l'usage des données correctement décodées.

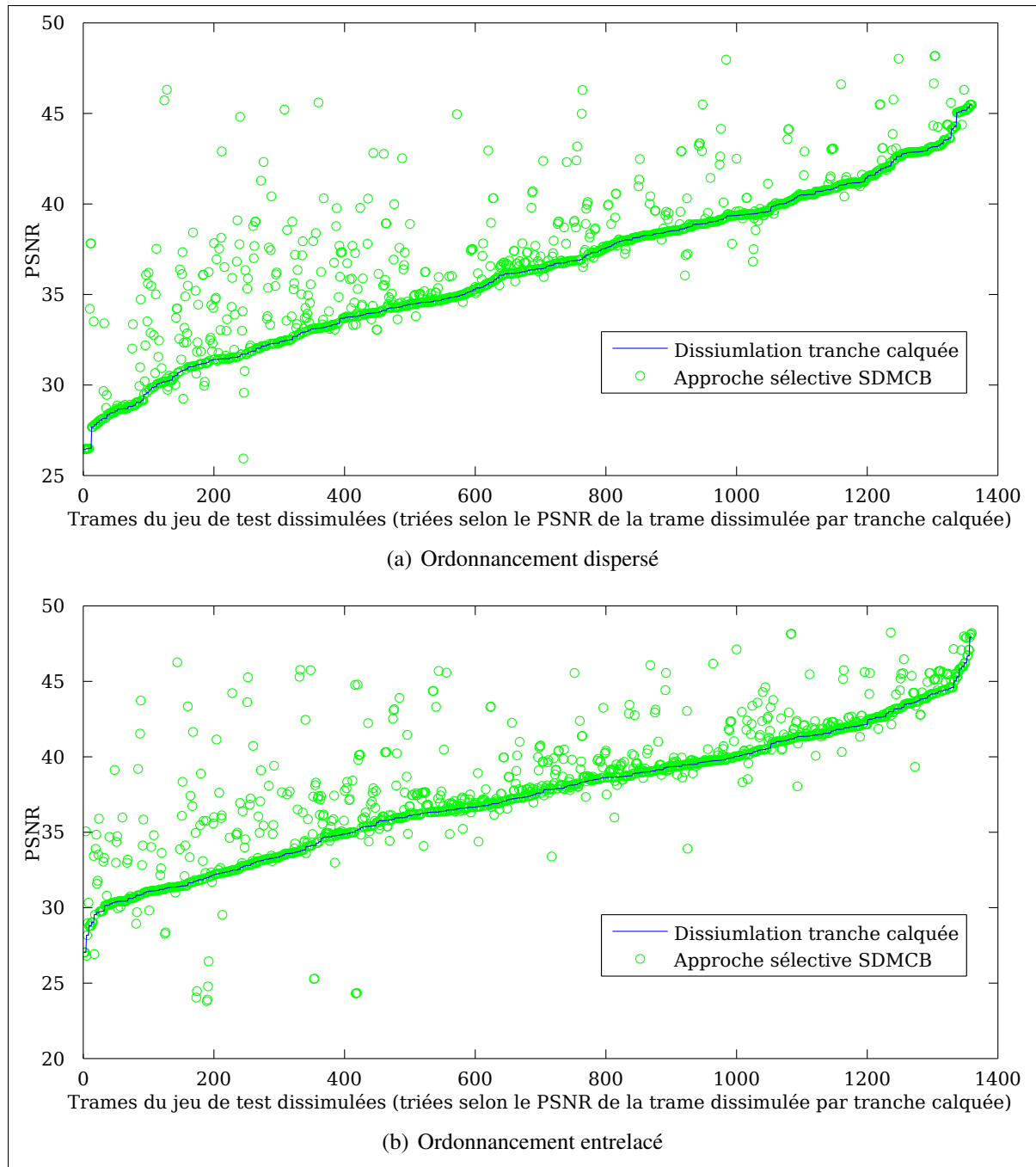


Figure 1.9 Visualisation de la distribution du PSNR de l'approche sélective basée sur SDMCB (ronds verts) à l'égard de la dissimulation par tranche calquée (ligne bleue).

Sans altérer la configuration de notre banc de test, nous mesurons l'aptitude du MCB à identifier le meilleur bloc candidat entre ceux provenant de la trame corrompue et ceux de la dissi-

mulation par tranche calquée. Pour ce faire, nous allons utiliser une taille de bloc $B = 16$ et un seuil d'effet de bloc $T_b = 5000$.

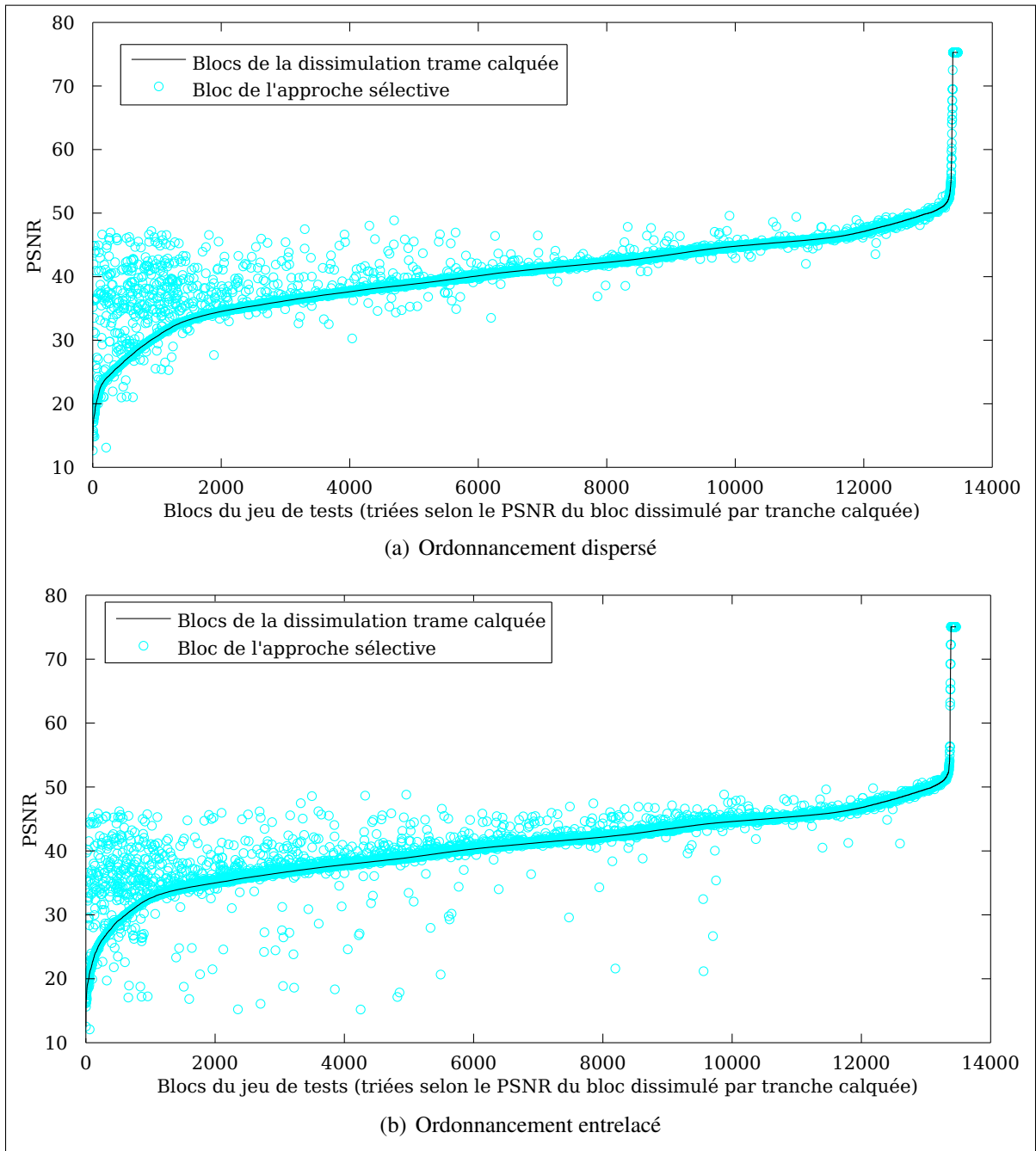


Figure 1.10 Visualisation de la distribution du PSNR issue des blocs résultants de l'approche sélective basée sur MCB (ronds verts) à l'égard de ceux produits par la dissimulation par tranche calquée (ligne bleue).

[SC]: vert ou turquoise/cyan ?

Les résultats obtenus sont présentés à la figure 1.10. La figure 1.10(a) démontre que l'approche sélective MCB choisit le bon bloc dans 88 % de cas. Ce qui produit un gain moyen de 0.86 dB sur l'ensemble des trames avec un ordonnancement de type dispersé. Pour l'ordonnancement entrelacé (figure 1.10(b)), l'approche sélective MCB effectue le bon choix dans 91 % des cas, permettant un gain moyen de 0.69 dB.

[SC]: pourquoi pas toutes les stats comme SDMCB somme le 1.98 dB et 1.24dB ? Est-ce par ce que tu ne sais plus à quel trame appartient chaque bloc ?

À l'aide de la figure 1.11, nous sommes en mesure d'analyser la distribution des PSNR résultats : de la trame erronée, de la dissimulation par tranche calquée et des approches sélectives. Cette figure expose la variation du nombre de trames selon des intervalles de 5 dB de PSNR. Ces intervalles s'étalent de 20 dB à 50 dB, permettant ainsi de visualiser la variation de la qualité visuelle issue de chaque approche. Commençons l'analyse en soulignant la présence d'une importante quantité de trames erronées dans les intervalles de faible qualité, où le PSNR est inférieur à 30 (les intervalles centrés autour de 20 et 25). Pour ces intervalles, nous constatons, autant dans l'histogramme de la figure 1.11(a) que 1.11(b), que la majorité de ces trames sont dissimulées par le calquage de tranche et les approches sélectives. Poursuivons l'analyse des approches sélectives avec le constat suivant, pour l'intervalle centré autour de 30 dB, le nombre de trames liées aux approches sélectives est inférieur à celui lié à la dissimulation par tranche calquée. Cette observation n'est pas due à une perte de qualité, mais bien à une augmentation de la qualité, car on retrouve ces trames dans les intervalles supérieurs. Pour ces intervalles, les approches sélectives produisent un plus grand nombre de trames que la dissimulation par tranche calquée. Grâce à ces observations, nous sommes en mesure de conclure que les approches sélectives produisent de meilleurs résultats en exploitant les trames erronées de haute qualité visuelle, tout en évitant les trames erronées de mauvaise qualité. En ce qui a trait à la comparaison des approches sélectives, l'approche sélective MCB offre un plus grand nombre de trames de qualité visuelle pour les intervalles de 35 dB et 40 dB vis-à-vis SDMCB. Cepen-

nant, cette dernière se reprend dans l'intervalle de 45 dB. Cette reprise est attribuée au fait que le remplacement de blocs sporadique dans une trame peut engendrer une perte de qualité causée par le mauvais arrimage des valeurs de pixels en frontière de bloc. Cette légère dégradation réduit le nombre de trames de très haute qualité. Toutefois à des niveaux de 40dB et plus, la différence de qualité est souvent imperceptible.

[SC]: Expliquer que le PSNR est saturé à une certaine valeur et dire laquelle

Tableau 1.1 Résumé du PSNR moyen obtenu par les diverses approches présentées dans cette section sur le jeu de tests.

Approche	PSNR Moyen dispersé (dB)	PSNR Moyen entrelacé (dB)
Encodée (sans erreur)	41.218222	41.200377
Sélective par bloc (avec référence)	37.689039	38.528986
Sélective par tranche (avec référence)	37.117712	38.082769
Sélective MCB	37.034840	37.951004
Sélective SDMCB	36.901414	37.911514
Dissimulation tranche calquée	36.179630	37.263033
Trame corrompue	35.068784	36.075526

[SC]: Il faut présenter et analyser le tableau 1.1. Il manque aussi une petite conclusion au chapitre

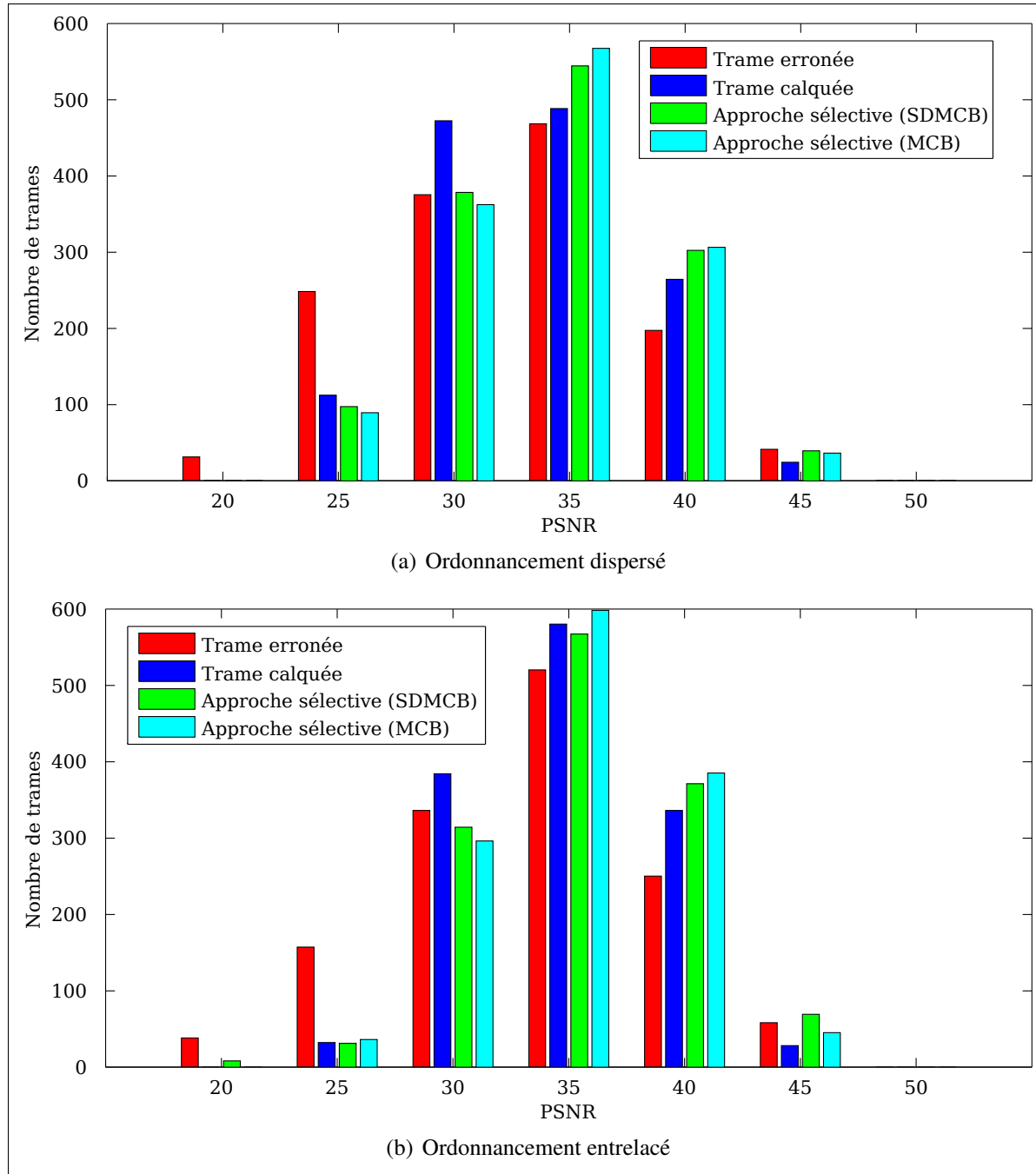


Figure 1.11 Histogrammes des PSNR des trames, avec des intervalles de 5 dB, centrés à chaque incrément de 5 dB, à partir de 20 dB. Les trames considérées sont celles résultantes : du décodage de la trame erronée (rouge), du calquage de tranche (bleu), de l'approche sélective par tranche (vert) ainsi que l'approche sélective par bloc (cyan).