

## CHAPITRE 1

### REVUE DE LA LITTÉRATURE

Dans ce chapitre, nous faisons état de la littérature sur les approches de détection de la détérioration visuelle. Il ne s'agit pas ici d'une simple énumération, mais bien, d'une suite logique d'ouvrages permettant de comprendre, l'introduction et l'évolution de plusieurs concepts fondamentaux, sur lesquels reposent notre contribution.

La détection de la détérioration visuelle est possible uniquement si le contenu de paquets corrompus peut être décodé. Pour ce faire, le décodeur du canal de transmissions doit interagir avec le décodeur vidéo (source).

[SC]: je ne comprends pas pourquoi cette interaction est requise. on peut décoder parfaitement s'il n'y a pas d'erreur et dans ce cas il n'y a pas d'interaction. De même pour les paquets perdus pour lesquels le décodeur applique du error concealment.

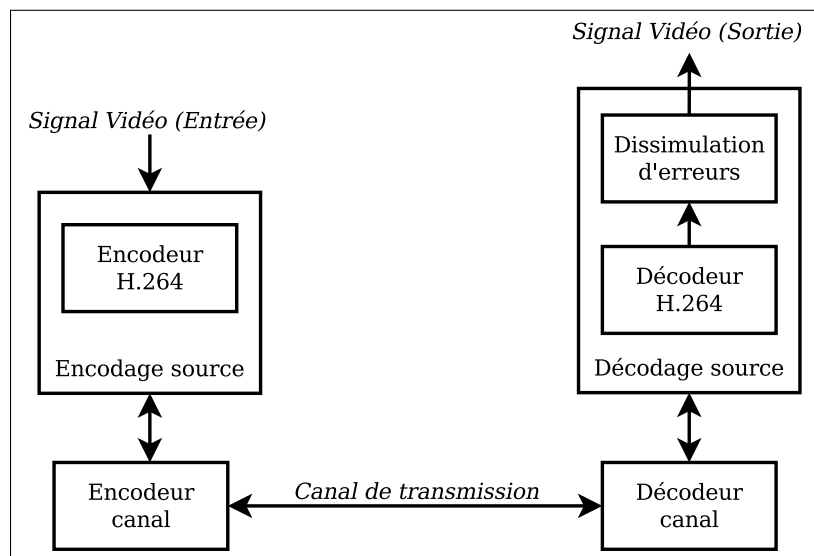
Ce type d'interaction porte le nom de *Joint Source Channel Decoding*. Notre prologue, section 1.1, expose les origines du *Joint Source Channel Decoding*. La section 1.2 offre un compte rendu des notions du *Joint Source Channel Decoding* qui s'appliquent spécifiquement au transport de séquences vidéo. Les améliorations apportées à une de ces notions, l'analyse syntaxique, sont présentées à la section 1.3. Finalement, à la section 1.4, les approches combinant l'analyse syntaxique à la détection de la détérioration visuelle dans le domaine des pixels sont décrites.

#### 1.1 Prologue

Dans son ouvrage *A Mathematical Theory of Communication*, ? explique sous l'allégorie ingénieuse du télégraphe, la notion même du *Joint Source Channel Decoding*. L'usage d'un langage redondant, tel l'anglais, comme encodage pour le télégraphe a l'avantage

d'être résistant aux erreurs du canal de transmissions. Le contexte du message permet la reconstruction des mots, même si plusieurs lettres de ceux-ci sont bruitées (?, p. 24).

En ?, ? sont les premières personnes à proposer d'utiliser le *Joint Source Channel Decoding* pour améliorer le transport d'images numériques. Déjà à cette époque, ? déplorent le gaspillage de bande passante introduit par les stratégies de résilience aux erreurs. Un gaspillage qui, 30 ans plus tard, est toujours présent sur les réseaux sans fil (?, p.1).



**Figure 1.1** Diagramme à blocs des composants liés au transport de séquences vidéo.  
Adaptée de ?, p. 976

## 1.2 Établir le lien entre le décodeur vidéo et celui du canal de transmission

Les techniques de résilience aux erreurs sur des réseaux non fiables ont suscité beaucoup d'intérêt dans les années 80 et 90. En ?, dans leur revue sur les techniques de contrôle et de dissimulation d'erreurs lors du transport de séquences vidéo (?), ? récapitulent, entre autres, l'état de l'art, de l'époque, des stratégies de résilience aux erreurs. ? présentent une multitude d'approches. Parmi celles-ci, ils identifient un type d'approches qu'ils qualifient d'approches de dissimulation par post-traitement effectué par le décodeur (?, Chap. 5). Ce type résume, de façon exacte, les approches que nous présentons dans cet ouvrage. Leurs composantes sont illustrées à la figure 1.1.

**[SC]:** expliquer chaque composante de la figure et en particulier le rôle du codage de source (enlever redondance), codage de canal (ajouter de la redondance pour augmenter la robustesse aux erreurs), dissimulation des erreurs au décodeur...

Parmi les approches de dissimulation présentées en [?], on remarque : *Motion-Compensated Temporal Prediction* ([?]), *Maximally Smooth Recovery* ([?]), *Projection onto Convex Sets (POCS)* ([?]), *Spatial- and Frequency-Domain, Interpolation* ([?]) et ([?]).

Toutes ces approches de post-traitement, effectuées par le décodeur, présentées par [?], supposent que les blocs endommagés sont connus. Pratiquement, cette identification est accomplie en ignorant les paquets corrompus, ce qui force le décodeur à dissimuler tous les blocs qui contiennent ces derniers même si un seul bloc est endommagé. Pour reprendre l'exemple du télégraphe, ceci revient à rejeter l'ensemble des mots d'une phrase qui possède une lettre en erreur. Cela complique grandement la compréhension du message. Ce nivellement par le bas, quoique plus simple, est un exemple du gaspillage identifié par [?], où le décodeur de la source ne communique pas avec le décodeur du canal de transport.

Le *Joint Source Channel Decoding* ([?]) cherche, à l'aide de la logique de la source, à identifier précisément, où se trouve l'erreur dans un ensemble de bits corrompus. Par exemple, pour l'encodage MPEG-4 Visual, [?] propose des règles permettant de vérifier la validité de : la portée des vecteurs de mouvement, des codes entropiques décodés, la portée des valeurs de coefficient DCT et du nombre de coefficients DCT afin qu'il ne dépasse pas le nombre permis par la norme ([?]). Une fois l'erreur identifiée, il existe deux types de solutions possibles : la reconstruction et la dissimulation.

D'une part, la reconstruction vise à retrouver les valeurs originales des bits corrompues à l'aide de modèles mathématiques complexes, comme présentée par [?].

D'autre part, la dissimulation tente de décoder les paquets corrompus, et par la suite, identifier et dissimuler la détérioration visuelle engendrée par cette opération. L'article de [?], parue en mai [?], présente un algorithme destiné au décodage d'images JPEG corrompues. Ce dernier est

capable d'identifier et classifier des blocs erronés, selon leurs caractéristiques dans le domaine des pixels (?).

? définissent quatre mesures permettant d'identifier les blocs erronés. La première est la validité de la valeur assignée à un pixel. Il s'agit ici de déterminer si un pixel est dans un intervalle de données plausibles. Par exemple : pour les valeurs d'un pixel non signées de niveaux de gris assignées sur huit bits, les valeurs possibles sont de 0 à 255. La deuxième mesure est l'hétérogénéité des bordures de blocs horizontales mesurée à l'aide d'un filtre Sobel. La troisième est l'hétérogénéité des bordures de blocs verticales, encore une fois mesurée à l'aide d'un filtre Sobel. Dans d'autres oeuvres, l'hétérogénéité en bordure de blocs est aussi connue sous le nom d'effet de bloc. Le filtre Sobel est défini par les opérateurs suivants (horizontal et vertical respectivement) :

$$S_h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_v = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (1.1)$$

La dernière mesure, présentée par ?, évalue la continuité des bordures propres au contenu de l'image qui traversent un bloc. Par la suite, ces mesures sont comparées à des seuils, afin de déterminer si le bloc évalué est erroné. Le résultat de la détection d'erreurs à l'aide de ces mesures est présenté à la figure 1.2, où les blocs noirs de l'image (b) représentent les erreurs détectées. On y constate de faux positifs, surtout dans le chapeau et les cheveux de Lena Söderberg.



**Figure 1.2** Exemple du résultat de la détection d’erreurs avec les mesures de ?  
Tirée de ?, p. 371

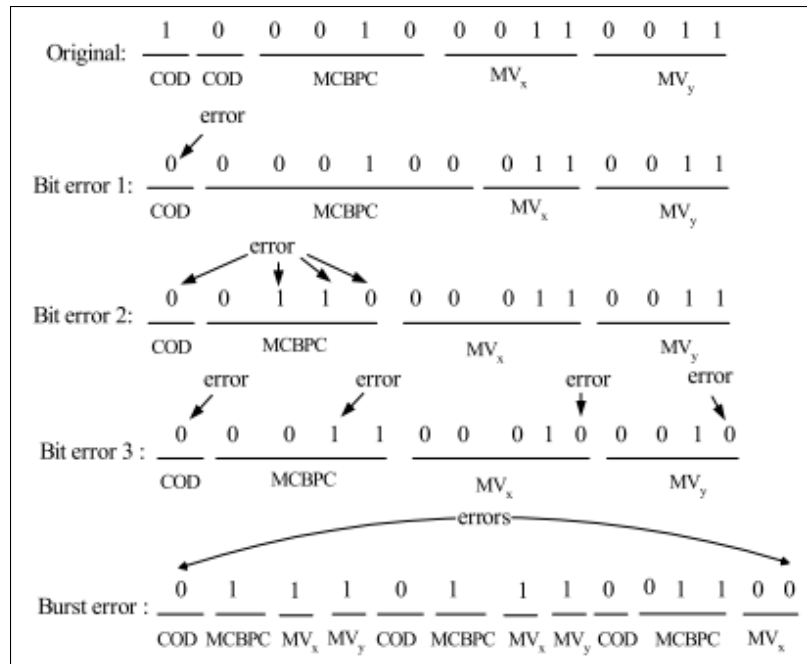
### 1.3 L’amélioration des approches d’analyse syntaxique d’encodage vidéo

Dans le but d’améliorer l’efficacité des règles de validation de syntaxe présentées par ?, ? proposent une nouvelle règle de validation qui augmente considérablement l’efficacité de ce genre d’approche. Cette règle permet de valider, entre deux marqueurs de synchronisation, que le nombre de COD

[SC]: définir l’acronyme COD : cash on delivery ;-)

soit égal au nombre de blocs (?). Expliquons le raisonnement derrière cette approche. Tout d’abord, les marqueurs de synchronisation sont des séquences binaires uniques qui servent de point de repère pour le décodeur. Le COD est un bit positionné en début de macrobloc, qui identifie si ce dernier est encodé ou pas. Le nombre de macroblocs contenus dans une tranche est disponible dans les paramètres d’encodages. S’il y a eu désynchronisation, ces deux nombres risquent d’être différents, comme illustré à la figure 1.3.

[SC]: Il faut présenter toutes les figures. Ce qu’elles présentent. Mais selon moi, l’explication du paragraphe suffit et on peut enlever la figure.



**Figure 1.3** L'impact de la corruption de bits, lors du décodage.

Adaptée de ?, p. 1425

Toujours dans le même ordre d'idée, en ?, ? proposent un valideur de syntaxe H.264. Ce valideur identifie des séquences de codes binaires invalides engendrées par la désynchronisation des codes entropiques, suite à une erreur (?).

**[SC]:** Ok mais quels résultats sont obtenus par leur approche ?

Ce valideur est exhaustif et définit des restrictions sur les valeurs d'un grand nombre de paramètres de la norme.

? proposent trois catégories d'erreurs de décodage selon les caractéristiques de l'erreur (?) :

- **Code numérique invalide** : Il s'agit ici d'un code qui n'est pas dans la table de codes numériques associés à ce paramètre.
- **Code hors de portée** : Lorsqu'une valeur décodée est à l'extérieur de l'ensemble des valeurs valides pour le paramètre en question.
- **Erreur contextuelle** : Cette erreur survient si le décodeur doit effectuer une opération invalide.

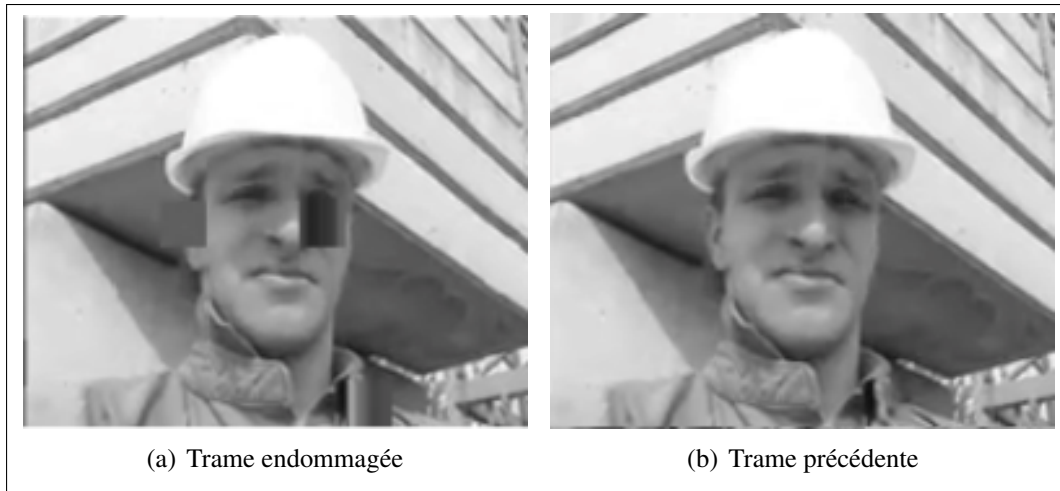
#### 1.4 La combinaison de l'analyse syntaxique et de la détection de la détérioration visuelle dans le domaine des pixels

En ?, ? combinent leur approche de validation de syntaxe à un algorithme d'identification de la détérioration visuelle inspiré de ?, auquel un système de vote a été ajouté, afin de raffiner le résultat de la détection (?). Dans son mémoire ?, explique en détail l'algorithme proposé (?).

La détection de la détérioration visuelle est réalisée dans le domaine des pixels. Pour isoler l'erreur, ? utilisent le différentiel de la trame à évaluer par rapport à celle qui la précède (c.-à-d. la différence entre ces deux trames). La figure 1.4 montre une trame endommagée ainsi que sa trame précédente. Le différentiel entre ces deux images est calculé en effectuant une différence entre les pixels correspondants (même position spatiale). Le résultat du différentiel des images de la figure 1.4 est présenté à la figure 1.5(a).

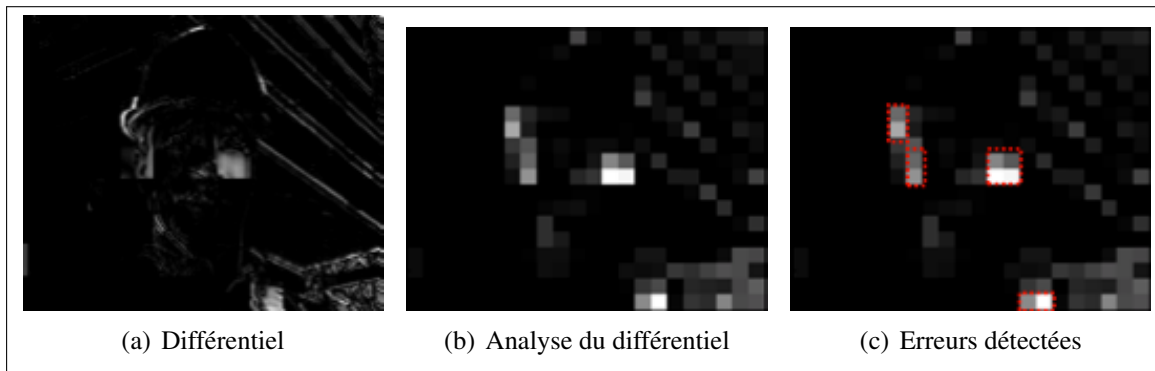
**[SC]:** Les bonnes figures 1.4 et 1.5 ? Toujours présenter l'information des figures. Le lecteur ne va pas comprendre ce que tu veux qu'il comprenne. Ici 1,4 n'est pas présenté

Par la suite, les valeurs du différentiel sont regroupées en blocs  $8 \times 8$  et la moyenne est assignée comme étant représentative de l'énergie du bloc (?, p. 27-28), comme illustré à la figure 1.5(b). Les erreurs détectées à la figure 1.5(c) sont obtenues à l'aide d'un seuil. Si la valeur de l'énergie d'un bloc est supérieure à ce seuil, le bloc est considéré comme erroné.



**Figure 1.4** Trames utilisées par ? pour démontrer l'algorithme de détection de la détérioration visuelle.

Tirée de ?, p. 25

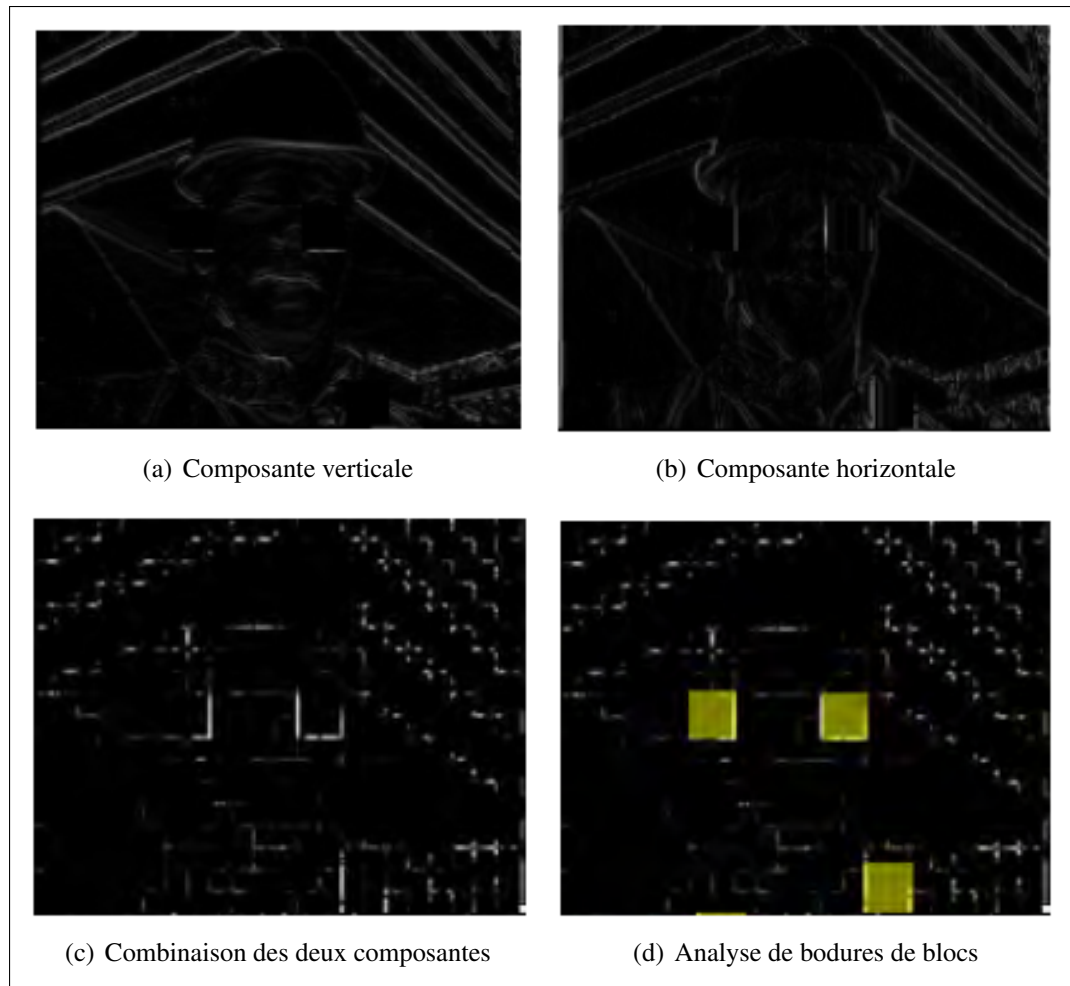


**Figure 1.5** Exemple de l'analyse et de la detection d'erreurs.

Tirée de ?, p. 25

De plus, la détection d'effets de bloc en bordure de blocs est effectuée. Pour trouver les bordures à l'intérieur des trames, les composantes horizontales 1.6(a) et verticales 1.6(b), d'un filtre de Haar (?), sont combinées et seulement les valeurs en bordure de bloc sont conservées 1.6(c). L'analyse des effets de bloc est accomplie pour les blocs  $8 \times 8$  et pour les macroblocs  $16 \times 16$ . Ceux-ci sont comparés, à des seuils, afin de savoir s'il y a présence d'erreurs.





**Figure 1.6** Composantes et analyse du filtre de Haar.

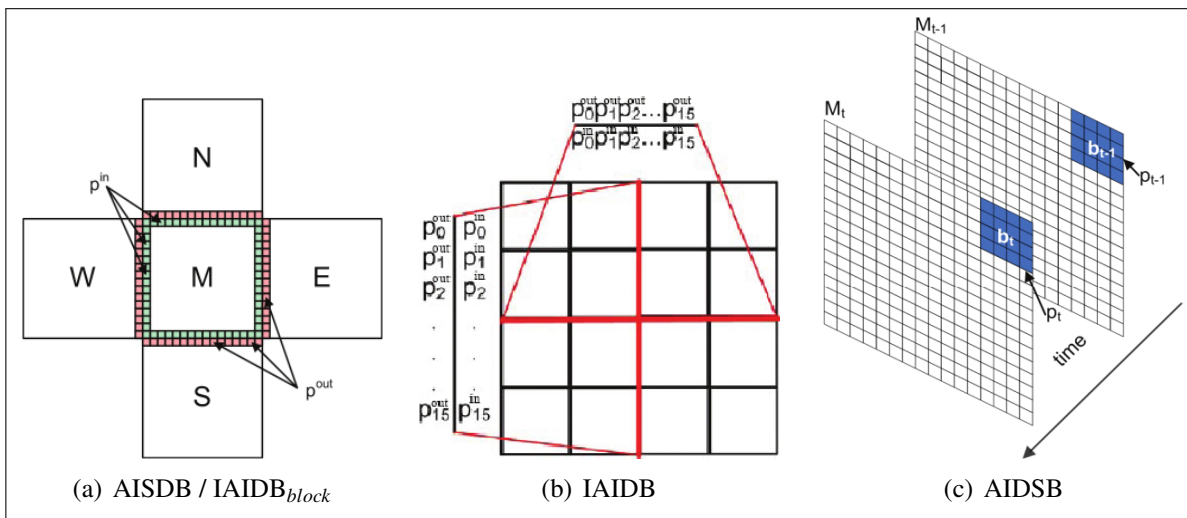
Tirée de ?, p. 26

Les mesures d'énergie et d'effets de bloc sont interprétées par un système de vote qui détermine où se trouvent les blocs erronés à l'intérieur de la trame évaluée.

L'année suivante, ? reprennent les travaux de ? et retirent les seuils et le système de vote. Ceux-ci sont remplacés par une machine à vecteurs de support (?). Cette dernière est une méthode d'apprentissage supervisée qui, une fois entraînée sur un ensemble de données, est capable de classifier d'autres données. Pour ce genre d'algorithme, le temps d'entraînement est long, mais la classification est très rapide, ce qui en fait un bon choix pour des considérations temps réel (?).

Le vecteur d'entrées (*feature vector*) défini par ? possède huit composantes :

1. AIDB
2. La moyenne du  $\text{IAIDB}_{\text{block}}$
3. L'écart type du  $\text{IAIDB}_{\text{block}}$
4. IAIDB vertical
5. IAIDB horizontal
6. La moyenne du AIDSB
7. L'écart type du AIDSB
8. TC



**Figure 1.7** Visualisation des composantes du vecteur d'entrées.

Tirée de ?, p. 79-80

**[SC]:** Expliciter tous les acronymes. Aussi la figure montre AISDB et tu parles de AIDB partout, c'est quoi la relation ?

Ce vecteur représente un macrobloc contenu à l'intérieur d'une trame d'une séquence vidéo. Le AIDB est la différence moyenne des pixels en bordure de macroblocs (M, à la figure 1.7(a)). Le  $\text{IAIDB}_{\text{block}}$  est la même mesure que le AIDB, mais appliquée aux blocs  $4 \times 4$  à l'intérieur d'un macrobloc. Le IAIDB est la moyenne de la différence des pixels pour les bordures

internes (verticales et horizontales) d'un macrobloc (lignes rouges figure 1.7(b)). Le AIDS<sub>B</sub> est l'énergie d'un macrobloc  $16 \times 16$ , tel que défini par ?, soit la moyenne du différentiel par rapport à la trame précédente (voir la figure 1.7(c)). La consistance de la texture (TC) est basée sur une approche d'analyse de texture présentée par ? connue sous le nom de patron binaire local (*Local Binary Pattern (LBP)*).

[SC]: Résultats de cette approche ? performace ?

[SC]: IMPORTANT : Dans un chapitre sur l'état de l'art, on s'attend à avoir une analyse critique des méthodes existantes avec leurs performances, leurs points forts et leurs points faibles. Cela sert à justifier pourquoi on choisit une approche différente.

[SC]: Revoir les références, il manque des accents à décembre. Revoir Ikunu 2007 car accent mal placé et 'master thesis' à franciser. REvoir avec la tite dame les question de référence et traduction de 'in proceeding' , 'in', ...

En somme, ce chapitre offre un bref survol des travaux de recherche en lien avec : le *Joint Source Channel Decoding*, la résilience aux erreurs lors du transport de vidéo, la validation de syntaxe pour les normes d'encodage vidéo et la détection de la détérioration visuelle dans le domaine des pixels. Ces concepts sont présentés dans un ordre quasi chronologique, où la cohésion conceptuelle a raison de la chronologie. Conséquemment, ils servent de notions fondatrices aux approches proposées dans cet ouvrage, et présentées dans le prochain chapitre.