

## **CHAPITRE 1**

### **UNE NOUVELLE APPROCHE D'IDENTIFICATION ET DE DISSIMULATION DE LA DÉTÉRIORATION VISUELLE**

Les chapitres antérieurs constituent les éléments d'un panorama technologique permettant l'assimilation des notions présentées dans ce chapitre. Ces éléments recouvrent un large territoire technologique s'étendant de l'encodage vidéo, au transport de celui-ci ainsi que l'impact, la détection et la dissimulation de la détérioration de séquences vidéos codées en respectant la norme H.264.

Rappelons d'abord, à la section 1.1, les limitations des approches actuelles de détection d'erreurs. Nous présentons ensuite, à la section 1.2, une amélioration fondamentale aux approches actuelles de détection de la dégradation visuelle à l'intérieur de trames corrompues. Cette approche de détection améliorée sert de base pour un nouveau type d'approche de dissimulation, les approches sélectives. À la section 1.3, deux variantes de celles-ci sont décrites.

#### **1.1 Les limitations des approches actuelles de détection de la détérioration visuelle**

Les approches de détection de la détérioration visuelle étudiées ici sont celles présentées lors de notre revue de la littérature. Avant de définir les limitations de ces approches, commençons par uniformiser la nomenclature utilisée pour décrire celles-ci. Définissons deux matrices :  $\mathbf{F}$ , celle contenant les pixels d'une trame d'une séquence vidéo et  $\mathbf{E}$  la matrice de pixels issue du décodage réussi des paquets de  $\mathbf{F}$ , corrompus lors du transport. Pour une matrice  $\mathbf{F}$ ,  $\mathbf{F}_{x,y}$  représente la valeur du pixel de la matrice à la position  $(x,y)$ .

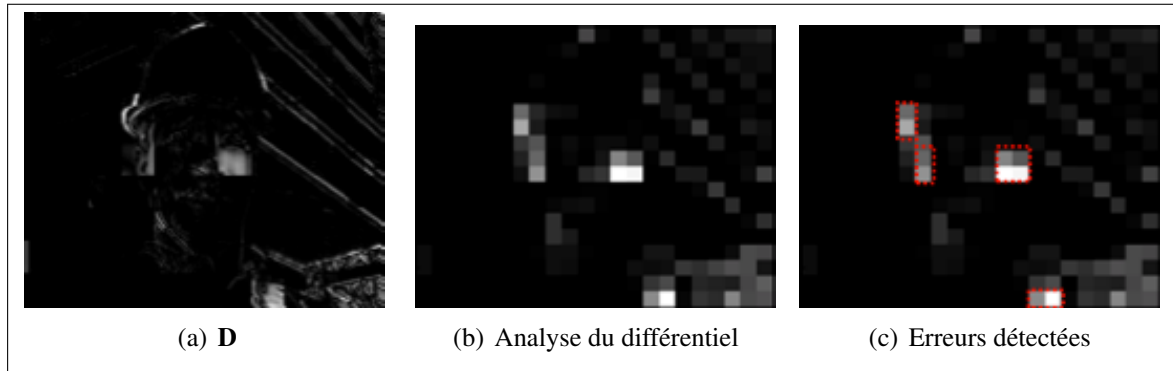
[SC]: On ne peut pas parler de paquets de  $\mathbf{F}$  car  $\mathbf{F}$  est une matrice. Selon moi il faudrait citer une figure de diagramme de système avec les matrices définies ici ainsi que les autres que tu définirais. Genre  $\mathbf{F}_O$  est ton image originale, elle est codée en  $\mathbf{F}$  et brisée en paquets  $p_i(\mathbf{F})$  qui sont corrompus et appelés  $\hat{p}_i(\mathbf{F})$  ou quelque chose du genre. Ici il faudrait aussi séparer ce qui est une notation (par exemple  $\mathbf{F}$  est une matrice et  $\mathbf{F}_{x,y}$  est la valeur à la position  $(x,y)$  est une notation générale) de la liste des signaux à définir :  $\mathbf{F}, \mathbf{E}, \mathbf{P}, \dots$ . Selon moi, c'est pour cela qu'une figure du système avec les signaux  $\mathbf{F}, \mathbf{E}, \mathbf{P}, \dots$  est requise ou au moins les définir. Pour la notation matrice pas besoin de définir pour  $\mathbf{F}$  et  $\mathbf{E}$  juste un des deux (ou une matrice générique  $\mathbf{F}$ ). Tu pourrais définir  $\mathbf{F}_O(t)$  la trame originale au temps  $t$  (pas compressée),  $\mathbf{F}(t)$  la trame encodée et décodée sans erreur au temps  $t$ ,  $\hat{\mathbf{F}}(t)$  la trame encodée, transmise, puis décodée au temps  $t$  (donc possiblement corrompue). Et tu dirais que pour alléger la notation, lorsque le contexte sera clair, on notera :  $\mathbf{F}$  pour  $\mathbf{F}(t)$  la trame courante,  $\mathbf{E}$  pour  $\hat{\mathbf{F}}(t)$  la trame courante erronée,  $\mathbf{P}$  pour  $\mathbf{F}(t-1)$  la trame précédente (car on fait l'hypothèse qu'elle est sans erreur), etc.

Les auteurs, Superiori et al. (2007) ainsi que Ikuno (2007), présentent une approche de détection où l'on mesure l'énergie des blocs ainsi que leurs effets de blocs à l'intérieur d'un différentiel  $\mathbf{D}$ . Ce différentiel, issue de la différence absolue entre la trame erronée  $\mathbf{E}$  et la trame qui la précède  $\mathbf{P}$ , est défini par la formule suivante :

$$\mathbf{D}_{x,y} = \underline{\quad} |\mathbf{E}_{x,y} - \mathbf{P}_{x,y}|, \quad (1.1)$$

Les données obtenues par ces mesures sont comparées à des seuils et un système de votes détermine s'il y a erreur. La méthode se base sur l'hypothèse que les pixels ne varient pas beaucoup d'une image à l'autre et que les grandes variations sont probablement dues à des erreurs de transmission. Nous présentons, à titre d'exemple de cette approche, la figure 1.1 tirée de Ikuno (2007, p. 25). Pour éviter le recours aux seuils, les auteurs Farrugia et Debono

(2008) collectent une série de mesures provenant de  $\mathbf{D}$  et utilisent une machine à vecteurs de support (Cortes et Vapnik, 1995) pour identifier les blocs erronés.



**Figure 1.1** Exemple de l'analyse et de la détection d'erreurs.  
Tirée de Ikuno (2007, p. 25)

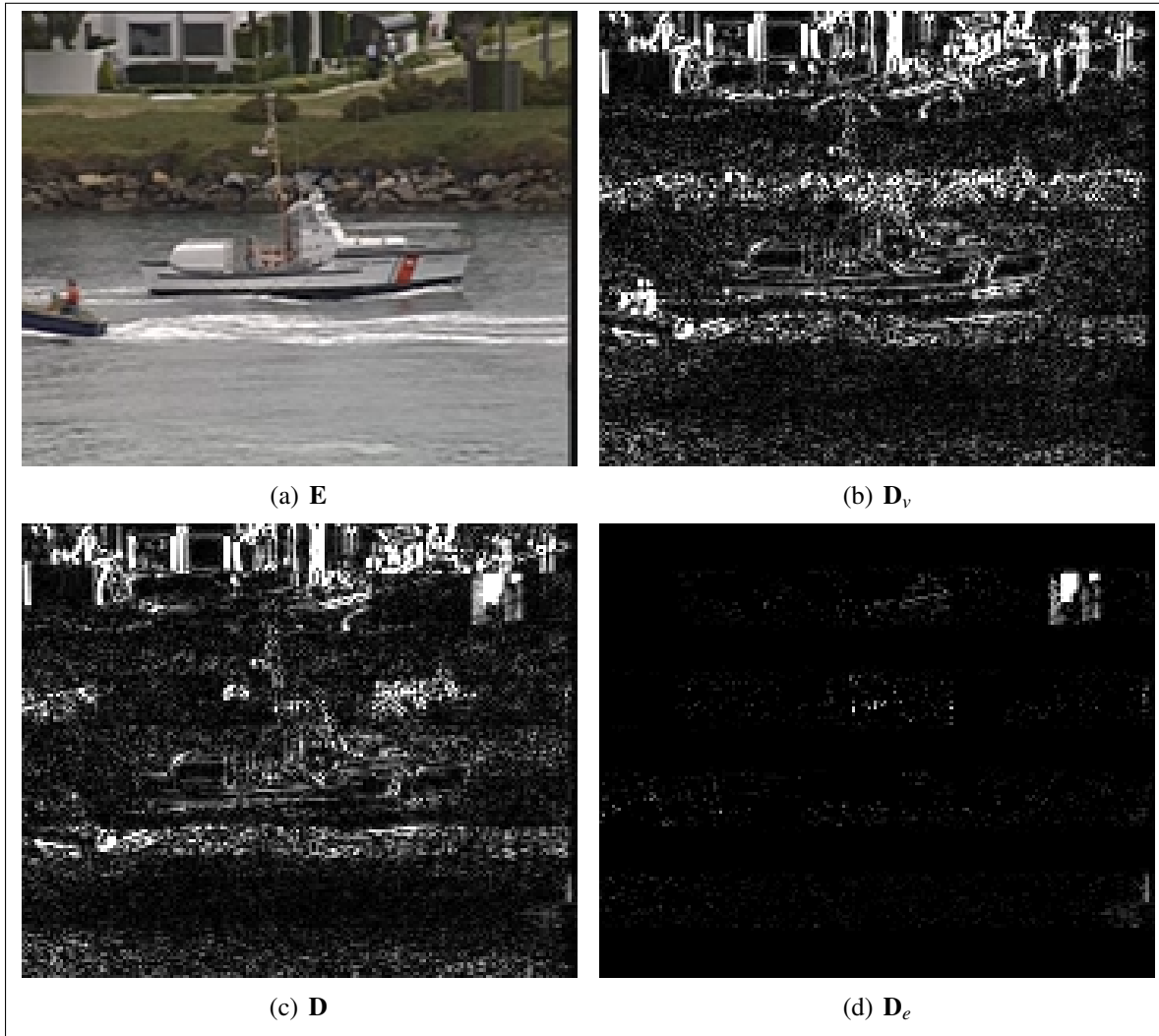
[SC]: Ce serait bien de voir l'image E et P

Le problème commun que nous identifions, comme limitation majeure, avec ces approches est l'usage du différentiel  $\mathbf{D}$ . Quoiqu'il puisse sembler alléchant, ce dernier, illustré à la figure 1.1(a), ne contient pas uniquement l'erreur due aux erreurs de transmission, mais aussi la variation des valeurs de pixels propre à la séquence. C'est-à-dire que le différentiel résultant des transformations entre les images non erronées, entre  $\mathbf{F}$  et  $\mathbf{P}$ , tel le mouvement des objets, est aussi contenu dans  $\mathbf{D}$ . En effet, utilisant l'inégalité du triangle ( $|a - b| \leq |a - c| + |c - b|$ ), on peut réécrire l'équation précédente comme :

$$\mathbf{D} = |\mathbf{E} - \mathbf{P}| \leq \underbrace{|\mathbf{E} - \mathbf{F}|}_{\mathbf{D}_e} + \underbrace{|\mathbf{F} - \mathbf{P}|}_{\mathbf{D}_v} \quad (1.2)$$

avec  $\mathbf{D}_e$  l'erreur produite par les détériorations visuelles due aux erreurs de transmission et  $\mathbf{D}_v$  l'erreur produite par les variations entre les trames. On peut remarquer que sans  $\mathbf{F}$ , il est impossible de distinguer, à l'intérieur de  $\mathbf{D}$ , les contributions individuelles de  $\mathbf{D}_e$  et de  $\mathbf{D}_v$ . Les auteurs mentionnés précédemment (Superiori et al. (2007), Ikuno (2007) ainsi que Farrugia et Debono (2008)) misent sur les caractéristiques spatiales de la dégradation visuelle pour tenter

d'effectuer cette distinction. Cependant, il est probable que la variation entre les trames  $\mathbf{D}_v$  manifeste ce genre de caractéristiques. Comme nous le démontrons aux figures 1.2(b) et 1.2(c).



**Figure 1.2** Contre-exemple de l'usage de  $\mathbf{D}$  pour effectuer la détection d'erreurs.

[SC]: peux-tu remplacer remplacer la figure d) pour avoir  $\mathbf{D}_e = |\mathbf{E} - \mathbf{F}|$  plutôt que  $\mathbf{D} - \mathbf{D}_v \leq \mathbf{D}_e$

Par notre contre-exemple, la figure 1.2, nous démontrons l'inefficacité de la détection d'erreurs par  $\mathbf{D}$ . La séquence *coastguard* est caractérisée par un déplacement horizontal de la caméra. Ce déplacement crée une très grande variation entre deux trames consécutives, comme illustré à la

figure 1.2(b). L'erreur à détecter est identifiée à la figure 1.2(d). Il n'est pas évident d'extraire cette information de **D** (figure 1.2(c)) sans connaître **F**.

Bref, ce manque de précision, engendré par la variation *intertrame* présente dans le différentiel, force les auteurs à employer des systèmes complexes de votes ou de machines à vecteur de support. À l'opposé, le résultat de notre effort de recherche est une approche de détection simple qui oeuvre dans un contexte différent de **D**, où la variation entre les trames est minimisée par la compensation de mouvement.

## 1.2 Les effets de blocs compensés par le mouvement

Nous venons de présenter la variation entre les trames comme un obstacle majeur à la détection d'erreurs. En s'intéressant aux origines de cette variation, on constate qu'elle peut provenir : d'une translation, d'une déformation, d'un changement intensité lumineux ainsi que de l'insertion ou du retrait de contenu.

**[LT]:** Faire le lien avec la section sur les erreurs dans le vidéo.

En pratique, l'erreur de transmission agit comme un changement entre les trames équivalant soit à une déformation ou une insertion de contenu. Ce qui fait en sorte que ces types de changements sont les plus difficiles à distinguer de l'erreur.

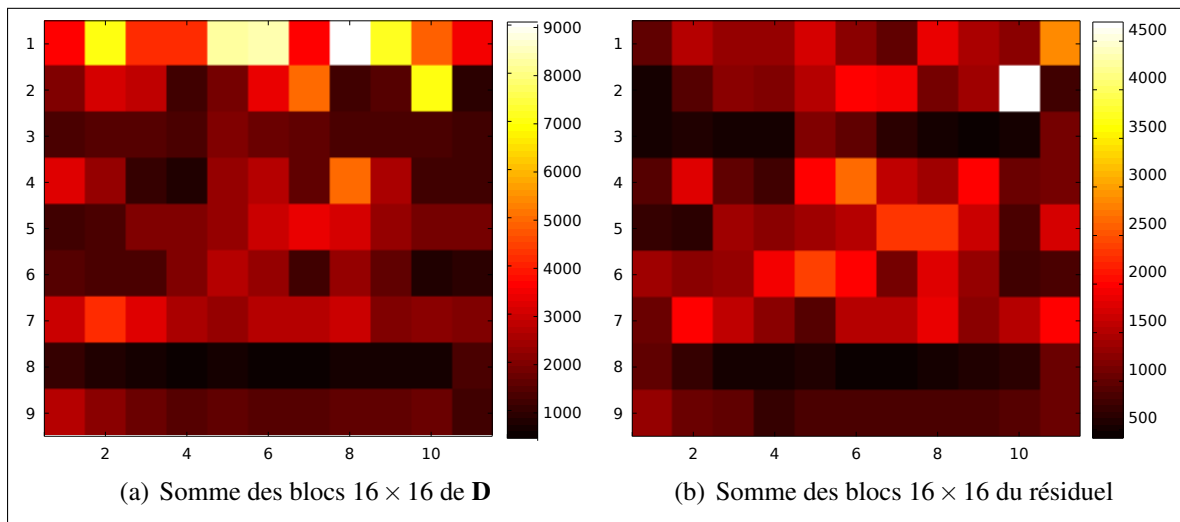
Dans le domaine de l'encodage vidéo numérique, la variation entre les trames est un concept qui a suscité une grande quantité d'efforts de recherche. Comme présentées dans notre section sur H.264, les approches de recherche et de compensation de mouvement sont raffinées, depuis maintenant plusieurs années, pour modéliser cette variation afin d'augmenter le taux de compression issue de leur encodage.

**[LT]:** Faire le lien avec la formule de recherche de mouvement du chapitre sur H.264

Dans notre contexte, ces algorithmes peuvent servir à prendre en compte les changements entre les trames au lieu de les ignorer, comme c'est le cas avec les approches qui reposent sur **D**. La

figure 1.3 illustre bien que l'erreur, présentée à la figure 1.2(d), est mieux identifiée par l'erreur de prédiction associée à la l'estimation du mouvement, nommée résiduel 1.3(b) (où seul le bloc erroné a une valeur supérieure à 3500) que par **D** 1.3(a) (où plusieurs blocs offrent des pointages excédant largement à celui de l'erreur).

**[SC]:** Le lecteur ne peut pas comprendre ni ce paragraphe ni cette figure. le caption est à revoir aussi. la figure montre rouge et rouge. le lecteur ne sera pas convaincu.



**Figure 1.3** Résultats de la sommation des pixels contenus dans les macroblocs de **D** (a) et du résiduel(b) sur la figure 1.2(a).

Quoiqu'elle améliore la détection d'erreurs, la compensation de mouvement n'est pas en mesure d'identifier les vecteurs de mouvement corrompus

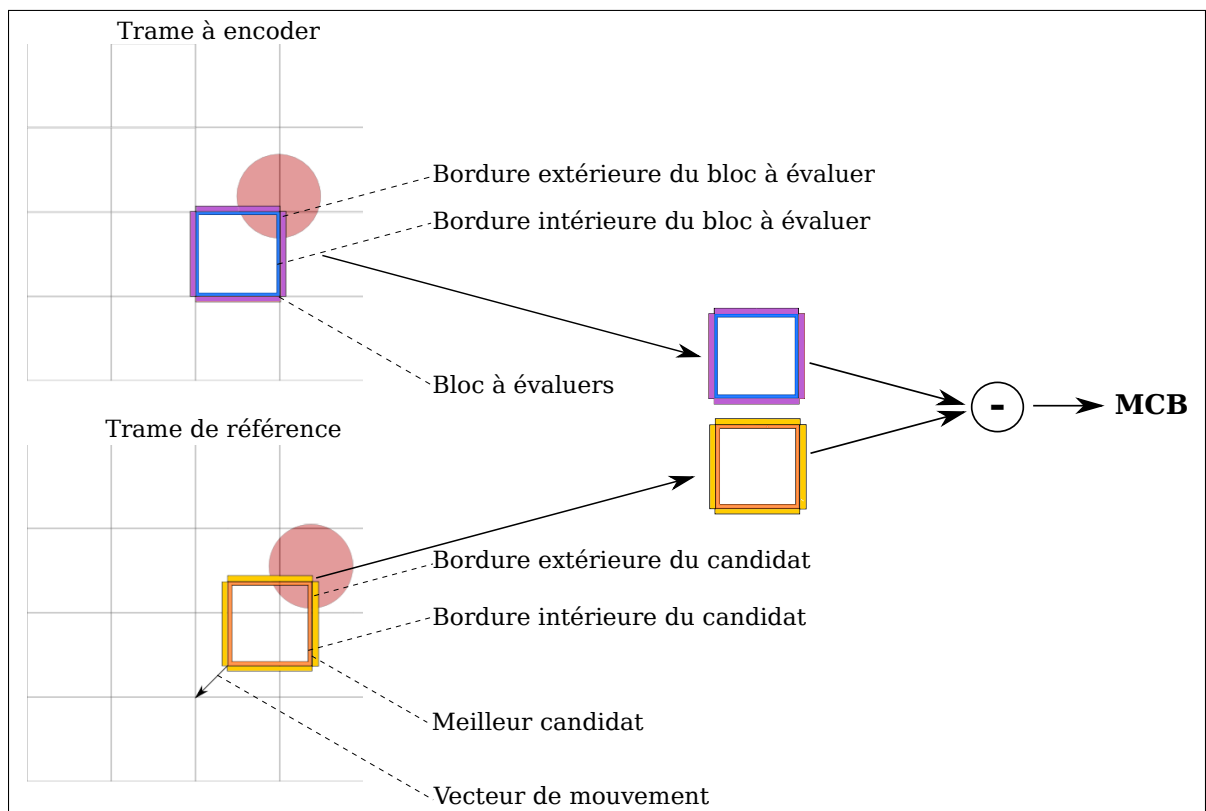
**[SC]:** Je suis perdu, la compensation du mouvement sert au codage. pourquoi on parle de compensation de mouvement en lien avec les VM corrompus ? est-ce que le lecteur a le background dans les autres chapitres ?

. Comme décrit au chapitre ??, ce type d'erreur insère du contenu fautif dans l'image erronée, provenant du mauvais endroit dans la trame de référence. Pour ce genre d'erreur, la recherche de mouvement retrouve le mauvais contenu et n'est pas en mesure d'identifier l'erreur.

[SC]: il trouve du contenu mais c'est le mauvais contenu qu'il utilise pour sa prédiction

[LT]: Ajouter référence sur le chapitre des erreurs.

Cette limitation de la recherche de mouvement nous apporte à incorporer un élément important pour la détection d'erreurs. La notion de nuisance. Ce que nous cherchons à détecter n'est pas l'erreur absolue, mais bien l'erreur qui engendre une dégradation visuelle perceptible par le système visuel humain. Dans le chapitre sur l'effet des erreurs sur les séquences vidéo, nous avons conclu que pour être dérangeantes, les erreurs doivent exhiber un ou plusieurs effets de blocs.



**Figure 1.4** Visualisation des composantes liées au calcul du MCB.

[SC]: changer évalueurs par evaluer. doit-on dire trame à encoder ou trame à décoder ?

Les effets de blocs compensés par le mouvement (MCB), que nous proposons dans cette recherche, sont la combinaison de ces deux faits. Comme décrit à la figure 1.4, le MCB recherche les discontinuités spatiales en bordure de blocs entre la trame à encoder et la trame de référence. Ne connaissant pas la trame de référence, nous utilisons la trame précédente.

Avant de procéder à l'explication pratique du fonctionnement du MCB (à la sous-section 1.2.2), nous présentons en détail, à la section 1.2.1, sa définition mathématique afin de bien comprendre comment le MCB est mesuré.

### 1.2.1 Définition théorique du MCB

Pour cette démonstration et à des fins de simplicité, nous avons recours à des blocs carrés<sup>1</sup> de taille  $B$ . Nous définissons un bloc comme un ensemble de pixels de taille  $B \times B$ . La hauteur et la largeur d'une trame sont définies, respectivement, par  $H$  et  $W$ . Les intervalles  $I_m = [0, \frac{W}{B} - 1]$  et  $I_n = [0, \frac{H}{B} - 1]$  représentent les indices de blocs à l'intérieur d'une trame. Nous désignons  $p$ , comme la demi-hauteur de la fenêtre de recherche de mouvement.

Nous employons  $(u, v)$ , comme le vecteur optimal issu de la recherche de mouvement. Cette opération s'exprime par la formule suivante :

$$(\mathbf{U}_{m,n}, \mathbf{V}_{m,n}) = \arg \min_{(u,v) \in K} \text{SAD}(mB, nB, u, v, \mathbf{E}, \mathbf{P}), \quad (1.3)$$

où  $K = [-p, p] \times [-p, p]$ ,  $m \in I_m$ ,  $n \in I_n$  et

$$\text{SAD}(x, y, u, v, \mathbf{E}, \mathbf{P}) = \sum_{q=0}^{B-1} \sum_{r=0}^{B-1} |\mathbf{E}_{x+q, y+r} - \mathbf{P}_{x+q+u, y+r+v}|. \quad (1.4)$$

Les composantes des vecteurs de mouvements optimaux  $(u, v)$  sont insérées à l'intérieur des matrices  $\mathbf{U}$  et  $\mathbf{V}$ , à la position correspondante à l'indice du bloc. Cette opération est effectuée sur l'ensemble des blocs qui composent la trame  $\mathbf{E}$ . Comme il a déjà été mentionné, nous effectuons la recherche de vecteurs de mouvement afin de tenir compte des variations issues

---

1. Il est aussi possible d'utiliser des blocs rectangulaires.



de  $\mathbf{D}_v$  afin de les séparer de la mesure de détérioration visuelle. Ainsi, au lieu de faire une différence entre les blocs situés à la même position spatiale, notre métrique effectuera la différence sur les blocs correspondants d'une trame à l'autre en considérant le mouvement de ceux-ci. De cette manière, nous réduisons l'effet du mouvement des objets entre les trames sur la mesure de différence.

**[SC]:** pixel entier, quart, demi pixel ?

Pour établir l'appariement des blocs entre la trame erronée et la trame de référence, nous avons recours à la somme de la différence absolue (SAD) entre deux blocs (eq.1.4). Quoiqu'il n'en résulte pas une identification exacte du mouvement, cette approche est peu complexe et permet l'identification du bloc candidat avec la plus faible variation par rapport au bloc désiré. Ainsi, lorsque, pour un bloc donné le SAD minimum est élevé, on peut souvent conclure qu'une erreur s'est produite.

Toutefois, l'usage du SAD minimum, tel que calculé en (1.3), comme approche de détection d'erreurs n'est pas efficace lorsque le contenu erroné est présent dans la trame précédente  $\mathbf{P}$  (c.-à-d. lorsqu'un bloc possède un bloc similaire dans la trame précédente). Cette situation se présente souvent lorsque les vecteurs de mouvement sont corrompus. Dans cette situation, ces vecteurs pointent sur le mauvais contenu même si le SAD est faible, ce qui peut souvent entraîner des effets de blocs importants.

C'est pour cette raison que, comme illustré à la figure 1.4, nous mesurons plutôt les effets de blocs en bordure du bloc à évaluer, par rapport aux effets de blocs du candidat, indiqué par le vecteur  $(u, v)$ , résultant de la recherche de mouvement (eq.1.3). Nous nommons cette opération : la mesure d'effets de bloc compensés par le mouvement. On y réfère aussi par son acronyme anglais MCB, et elle est définie par la formule :

$$\text{MCB}_d(\mathbf{E}, \mathbf{P}, m, n) = \sum_{l=0}^{B-1} |\mathbf{b}_{d,l}(\mathbf{E}, mB, nB) - \mathbf{b}_{d,l}(\mathbf{P}, mB + \mathbf{U}_{m,n}, nB + \mathbf{V}_{m,n})|, \quad (1.5)$$

où  $m \in I_m$ ,  $n \in I_n$ . De plus,  $\mathbf{U}$  et  $\mathbf{V}$  proviennent de l'équation (1.3). Le MCB est appliqué à toutes les bordures d'un bloc ( $d \in \mathcal{B}$ ). Nous référons aux bordures d'un bloc par l'ensemble suivant :  $\mathcal{B} = \{\mathcal{N}, \mathcal{E}, \mathcal{S}, \mathcal{W}\}$  et nous mesurons les vecteurs d'effets de bloc  $\mathbf{b}_{d,l}$  pour chacune d'elles, à l'aide de :

$$\mathbf{b}_{\mathcal{N},l}(\mathbf{F}, x, y) = \mathbf{F}_{x+l,y} - \mathbf{F}_{x+l,y-1}, \quad (1.6)$$

$$\mathbf{b}_{\mathcal{E},l}(\mathbf{F}, x, y) = \mathbf{F}_{x+B,y+l} - \mathbf{F}_{x+B-1,y+l}, \quad (1.7)$$

$$\mathbf{b}_{\mathcal{S},l}(\mathbf{F}, x, y) = \mathbf{F}_{x+l,y+B} - \mathbf{F}_{x+l,y+B-1}, \quad (1.8)$$

$$\mathbf{b}_{\mathcal{W},l}(\mathbf{F}, x, y) = \mathbf{F}_{x,y+l} - \mathbf{F}_{x-1,y+l}, \quad (1.9)$$

où  $l \in [0, B-1]$ , et  $(x, y)$  sont les coordonnées en pixels des blocs.

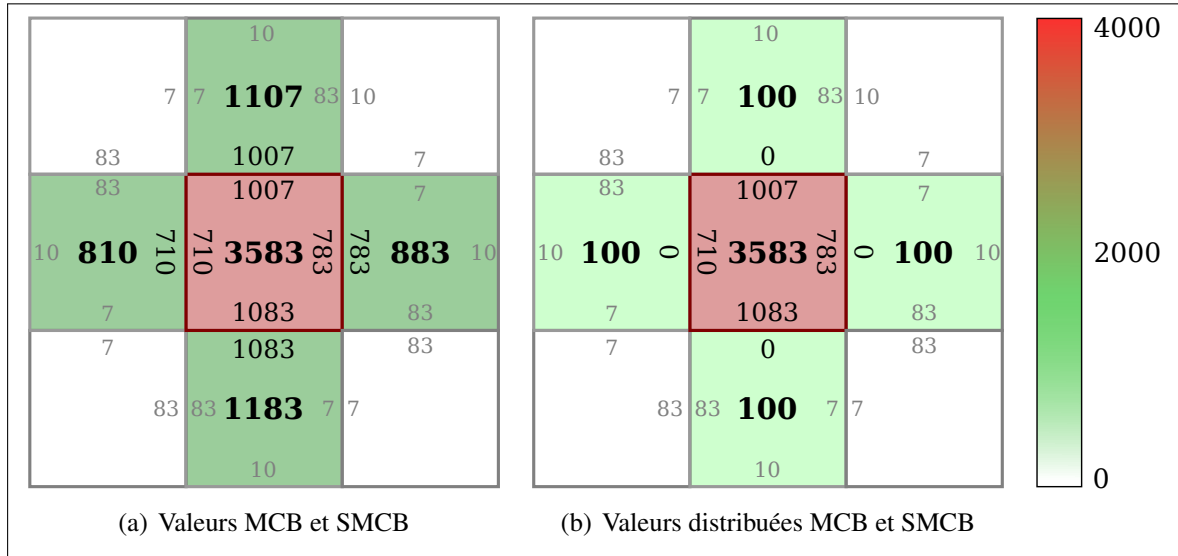
Pour évaluer l'effet global des effets de blocs en bordure d'un bloc, nous utilisons la somme des effets de blocs compensés par le mouvement de ce dernier (SMCB). Cette mesure est obtenue, pour un bloc aux coordonnées de bloc  $(m, n)$ , à l'aide de la formule suivante :

$$\text{SMCB}(\mathbf{F}, \mathbf{P}, m, n) = \sum_{d \in \mathcal{B}} \text{MCB}_d(\mathbf{F}, \mathbf{P}, m, n). \quad (1.10)$$

Dans cette formule, on effectue la sommation des valeurs MCB pour chacune des bordures  $d \in \mathcal{B}$  du bloc évalué. Ce qui est intéressant avec cette nouvelle approche c'est que nous mesurons le niveau de discontinuité présent à la bordure du bloc calculé par la soustraction du bloc considéré et le bloc qui lui correspond le mieux dans l'image précédente. S'il n'y a aucun bloc correspondant bien ou si le bloc correspond bien mais provient de l'application d'un mauvais vecteur de mouvement et qu'il est ainsi hors de son contexte initial, la mesure sera élevée. Sinon, si un objet s'est seulement déplacé et ainsi le contexte visuel est le même autour du bloc considéré et son bloc correspondant, et ce même s'il y a beaucoup de détails ou des bordures marquées autour de ces blocs, alors la mesure sera faible par l'effet soustractif de la mesure.

[SC]: une figure pour illustrée ces cas ?

La détection d'effets de bloc par la mesure de pixels en bordure de blocs peut toutefois introduire de fausses détections pour les bordures de blocs avoisinant un bloc erroné. Comme illustré à la figure 1.5(a), dans une telle situation, la valeur du SMCB des blocs voisins à la détérioration visuelle est considérablement augmentée par la bordure limitrophe à l'erreur.



**Figure 1.5** Valeurs MCB des bordures et valeurs SMCB (au centre) des blocs avant (a) et après (b) la distribution de bordures.

Pour résoudre ce problème, nous proposons le SDMCB, soit une version du SMCB où les bordures à valeurs élevées sont distribuées. La distribution de bordures est guidée par la valeur du SMCB. La valeur d'une bordure élevée est assignée au bloc avec le plus grand SMCB. La figure 1.5 montre le résultat de la distribution et l'algorithme 1.1 démontre comment celle-ci est effectuée.

L'algorithme de distribution de bordures 1.1 est appliqué, en ordre décroissant de valeur de SMCB, sur tous les blocs dont la valeur de SMCB est supérieure à un seuil  $T_b$ . Le recours au seuil est justifié, car si la valeur du SMCB est trop basse, la distribution n'est pas précise et ne sert à rien. Le but de la distribution est d'éviter la propagation d'effet de blocs importants aux blocs voisins. Le seuil  $T_b$  établit la valeur à partir de laquelle un bloc est considéré comme ayant des effets de blocs importants.

---

**Algorithme 1.1** Distribution de bordures
 

---

```

%Bordure supérieure
si  $\text{SMCB}(\mathbf{F}, \mathbf{P}, m, n) < \text{SMCB}(\mathbf{F}, \mathbf{P}, m, n - 1)$  alors
     $\text{MCB}_{\mathcal{N}}(\mathbf{F}, \mathbf{P}, m, n) = 0$ 
sinon
     $\text{MCB}_{\mathcal{S}}(\mathbf{F}, \mathbf{P}, m, n - 1) = 0$ 
fin si

%Bordure droite
si  $\text{SMCB}(\mathbf{F}, \mathbf{P}, m, n) < \text{SMCB}(\mathbf{F}, \mathbf{P}, m + 1, n)$  alors
     $\text{MCB}_{\mathcal{E}}(\mathbf{F}, \mathbf{P}, m, n) = 0$ 
sinon
     $\text{MCB}_{\mathcal{W}}(\mathbf{F}, \mathbf{P}, m + 1, n) = 0$ 
fin si

%Bordure inférieure
si  $\text{SMCB}(\mathbf{F}, \mathbf{P}, m, n) < \text{SMCB}(\mathbf{F}, \mathbf{P}, m, n + 1)$  alors
     $\text{MCB}_{\mathcal{S}}(\mathbf{F}, \mathbf{P}, m, n) = 0$ 
sinon
     $\text{MCB}_{\mathcal{N}}(\mathbf{F}, \mathbf{P}, m, n + 1) = 0$ 
fin si

%Bordure gauche
si  $\text{SMCB}(\mathbf{F}, \mathbf{P}, m, n) < \text{SMCB}(\mathbf{F}, \mathbf{P}, m - 1, n)$  alors
     $\text{MCB}_{\mathcal{W}}(\mathbf{F}, \mathbf{P}, m, n) = 0$ 
sinon
     $\text{MCB}_{\mathcal{E}}(\mathbf{F}, \mathbf{P}, m - 1, n) = 0$ 
fin si
  
```

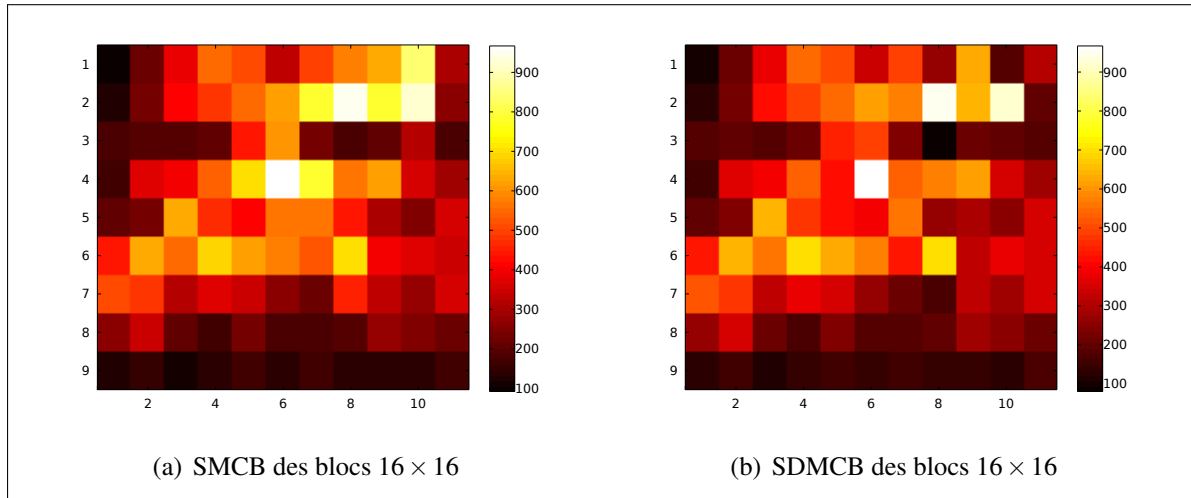
---

**1.2.2 Explication de l'approche MCB**

L'approche MCB comporte deux composantes clés pour l'identification d'erreurs. La première est spatiale, l'analyse de bordures identifie les discontinuités entre les pixels en bordure de blocs. Toutefois, ces discontinuités pourraient appartenir à l'image. La seconde composante est temporelle et, dans ce cas, recherche ces discontinuités dans la trame précédente.

Le seul cas problématique est lorsqu'il y a insertion ou déformation de contenu causant un effet de bloc important. Ce cas est rare et l'effet de bloc doit être considérablement important pour

être considéré comme erroné. Dans la prochaine section, nous présentons notre approche pour détecter des erreurs à l'aide des valeurs de MCB.



**Figure 1.6** Approche de détection d'erreur basées sur MCB pour la figure 1.2(a). Pour l'approche SDMCB un seuil  $T_b = 900$  a été utilisé.

Nous présentons, à la figure 1.6, les résultats obtenus suite à l'utilisation des approches SMCB et SDMCB sur la trame erronée de la figure 1.2(a). On y constate en (b) l'élimination de la propagation des valeurs élevées autour des blocs en erreurs (c.-à-d.  $SMCB > 800$ ). Lorsque comparé avec l'erreur présente dans la trame 1.2(d), nous sommes en mesure de conclure que l'approche est supérieure à **D** (voir figure 1.3(a)). Pour ce qui est du résiduel, c'est moins évident. Cependant, l'approche SDMCB 1.6(b) détecte non seulement l'erreur prédominante de la trame, mais aussi d'autres dégradations plus légères qui sont ignorées par le résiduel 1.3(b).

[SC]: pour moi le paragraphe n'est pas très convainquant. c'est difficile de comparer 1.3a, et 1.3b avec 1.6. Devrait-on dire que 3 spots blanc sont identifiés dans 1.6 et que l'un d'entre eux correspond à l'erreur? ou bien que 1.3a identifie un bloc blanc en haut qui n'a pas rapport... Mais c'est difficile de parler de couleurs si le mémoire est imprimé et tons de gris.

### 1.3 Les approches sélectives de dissimulation de la détérioration visuelle.

Dans cette section, nous sortons du contexte théorique de la détection d'erreurs et nous offrons une solution pratique de l'application du MCB. Celle-ci a pour but d'améliorer la qualité visuelle des dissimulations effectuées par le décodeur de référence H.264. Dans le cadre de nos recherches, nous avons intégré notre solution au décodeur inclus dans le logiciel de référence H.264/AVC JM, mais il est aussi concevable de le voir comme un module d'extension (*plugiciel*) destiné à d'autres décodeurs.

L'approche MCB assigne une valeur numérique à un bloc évalué en fonction de l'écart de son effet de bloc par rapport au bloc qui lui ressemble le plus dans la trame précédente. Nous affirmons, résultats à l'appui<sup>2</sup>, que cette approche permet d'identifier les détériorations visuelles importantes dans une trame corrompue. La quantification de la notion d'une dégradation visuelle importante a été un problème colossal que nous avons eu à surmonter, lors de notre implémentation pratique.

Une solution simple est l'usage d'un seuil. Cette valeur numérique définit un point après lequel un bloc avec un certain SDMCB est considéré comme erroné. Ce type d'approche, employée par Superiori et al. (2007) ainsi que Ikuno (2007), ne tient pas compte du caractère variant du contenu d'une séquence vidéo.

L'effet de bloc est relatif à l'image dans laquelle il est perçu. Ceci implique que l'usage d'un seuil fixe ne sera pas en mesure d'identifier les erreurs dans des séquences où les caractéristiques spatiotemporelles divergent.

De cette observation est née l'approche sélective. Comme son nom l'identifie, cette approche repose sur la notion d'un choix. Ce changement de paradigme provient du constat suivant : dans un contexte pratique, la dissimulation est inutile si l'erreur ne peut être dissimulée.

**[SC]:** Phrase bizarre. la détection est inutile ?

---

2. Voir le chapitre Résultats et analyse.

En se basant sur le fait que nous cherchons à effectuer une dissimulation, nous prenons d'un côté une dissimulation traditionnelle, telle la trame calquée ou le calque des vecteurs de mouvement et nous la comparons à la trame à évaluer. Par la suite, nous choisissons la plus petite valeur SDMCB. Que ce soit pour une trame, comme à la section 1.3.1, ou pour un macrobloc, section 1.3.2.

Ces approches se comportent comme des seuils dynamiques qui s'adaptent non seulement au contenu, mais aussi à l'aptitude dissimulation de la détérioration visuelle.

[SC]: Phrase bizarre. Revoir à l'aptitude dissimulation de la détérioration

En effectuant une dissimulation seulement lorsque du contenu de qualité supérieure (jugée par le SDMCB) est disponible.

[SC]: Phrase bizarre. Revoir le français de la phrase.

Une considération pratique importante est le temps de calcul requis par notre algorithme. Ce temps est restreint, vu les contraintes temps réel ainsi que les limitations matérielles des appareils mobiles. Ce qui est dispendieux en terme de temps de calcul pour le MCB est la recherche des vecteurs de mouvement. Cependant, les récentes avancées dans ce domaine, telles PMVFAST (Tourapis et al., 2001) et UMHExagonS (Cai et al., 2009) réduisent de façon substantielle le temps de calcul requis pour trouver les vecteurs de mouvement. De plus, notons que dans ce contexte, le décodeur est informé par les en-têtes RTP des paquets corrompus et mesure le SDMCB uniquement pour les trames qui en résultent.

### 1.3.1 L'approche sélective à l'échelon de la trame

[SC]: à l'échel ?

Notre première approche sélective repose sur le choix d'une trame. Lorsque des trames sont corrompues, le décodeur peut recourir à plusieurs trames pour la dissimulation : la tranche calquée, la tranche issue du calque des vecteurs de mouvement, la trame endommagée.

Le pointage SDMCB d'une trame est obtenu par la sommation des valeurs du SDMCB pour chacun de ses macrobloc. Cette sommation est décrite par la formule :

$$c_c = \sum_{m=0}^{\frac{W}{B}-1} \sum_{n=0}^{\frac{H}{B}-1} \text{SDMCB}(\mathbf{F}', \mathbf{P}, m, n), \quad (1.11)$$

où  $\mathbf{F}'$  est une trame dissimulée par le décodeur, soit par tranche calquée ou par la tranche issue du calque des vecteurs de mouvement, etc. On mesure une seconde sommation des valeurs SDMCB pour chaque bloc d'une trame. Cette fois, nous utilisons la trame endommagée  $\mathbf{E}$  :

$$c_e = \sum_{m=0}^{\frac{W}{B}-1} \sum_{n=0}^{\frac{H}{B}-1} \text{SDMCB}(\mathbf{E}, \mathbf{P}, m, n), \quad (1.12)$$

Par la suite, la trame avec le plus petit pointage est choisie comme dissimulation  $\mathbf{F}^*$  :

$$\mathbf{F}^* = \begin{cases} \mathbf{E}, & \text{si } c_e < c_c \\ \mathbf{F}', & \text{sinon} \end{cases}. \quad (1.13)$$

### 1.3.2 L'approche sélective à l'échelon du macrobloc

**[SC]:** à l'échel ?

Notre deuxième approche offre un niveau de granularité supérieur en effectuant la sélection pour chaque macrobloc. Le but de cette approche est de tirer profit du gain mutuel de différentes approches en fonction des caractéristiques spatiotemporelles des régions d'images.

Dans certaines conditions, la tranche calquée est optimale, dans d'autres c'est la trame erronée ou bien le calque des vecteurs de mouvement. De cette approche résulte une trame de dissimulation formée d'un conglomérat de macroblocs optimaux (jugés selon le SDMCB) issus de différents macroblocs candidats pour la dissimulation.



La dissimulation résultant de la sélection entre les blocs aux indices  $(m, n)$  provenant de la trame corrompue  $\mathbf{E}$  et la trame issue de la dissimulation par calquage de tranches  $\mathbf{F}'$  est obtenue à l'aide de

$$\mathbf{F}^*_{x,y} = \begin{cases} \mathbf{E}_{x,y}, & \text{si } \text{SDMCB}(\mathbf{E}, \mathbf{P}, m, n) < \text{SDMCB}(\mathbf{F}', \mathbf{P}, m, n) \\ \mathbf{F}'_{x,y}, & \text{sinon} \end{cases}, \quad (1.14)$$

pour tout  $x \in mB + [0, B - 1]$ , tout  $y \in nB + [0, B - 1]$ , et pour tout bloc de l'image (c.-à-d.  $m \in I_m = [0, \frac{W}{B} - 1]$ , et  $n \in I_n = [0, \frac{H}{B} - 1]$ ).

Dans ce chapitre, nous avons présenté deux variantes d'un nouveau type d'algorithme sélectif apte à la dissimulation d'erreurs. Ces variantes agissent à deux différents échelons de la hiérarchie de la vidéo numérique, soit ceux des macroblocs et des trames. Pour guider la sélection, nous proposons un algorithme de détection d'erreurs fondé sur une nouvelle notion : les effets de bloc compensés par le mouvement. Cette dernière prend en compte une partie des changements *intertrame*. Ceux-ci, comme nous avons démontré, entravent à la détection des algorithmes existants.

Le chapitre subséquent porte sur l'analyse des résultats expérimentaux obtenus à l'aide de notre banc d'essai, conçu pour valider les notions présentées dans ce chapitre.



## BIBLIOGRAPHIE

- Cai, C., Huanqiang Zeng, et Sanjit K. Mitra, Septembre 2009. Fast motion estimation for H.264. *Image Commun.*, 24 :630–636. ISSN 0923-5965. doi : 10.1016/j.image.2009.02.012. URL <http://portal.acm.org/citation.cfm?id=1595898.1596124>.
- Cortes, C. et Vladimir Vapnik, 1995. Support-vector networks. *Machine Learning*, 20 : 273–297. ISSN 0885-6125. URL <http://dx.doi.org/10.1007/BF00994018>. 10.1007/BF00994018.
- Farrugia, R. A. et Carl James Debono, Decembre 2008. A Robust Error Detection Mechanism for H.264/AVC Coded Video Sequences Based on Support Vector Machines. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(12) :1766–1770. ISSN 1051-8215. doi : 10.1109/TCSVT.2008.2004919.
- Ikuno, J. C., Decembre 2007. Performance of an Error Detection Mechanism for Damaged H.264/AVC Sequences. Master's thesis.
- Superiori, L., Olivia Nemethova, et Markus Rupp, Novembre 2007. Detection of visual impairments in the pixel domain of corrupted H.264/AVC packets. *Procs. of the Picture Coding Symposium (PCS 2007), Lisboa, Portugal*, pages 7–9.
- Tourapis, A. M., Oscar C. Au, et Ming L. Liou, 2001. Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) - Enhancing Block Based Motion Estimation. *Proceedings of Visual Communications and Image Processing 2001 (VCIP'01)*. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.9457>.