



Compression d'images et de vidéo: pourquoi, comment ?

Aline ROUMY

Chercheur, INRIA Rennes

SOMMAIRE

- 1.** Pourquoi compresser une vidéo ?
- 2.** Comment compresser ?
- 3.** Compression avec pertes
- 4.** Compression par transformée
- 5.** Compression par prédiction

1

Pourquoi compresser une vidéo ?

Définition

Compression = stocker même image avec moins de bits

Standards de compression : fournis par

ITU International Telecommunication Union

ISO International Organization for Standardization

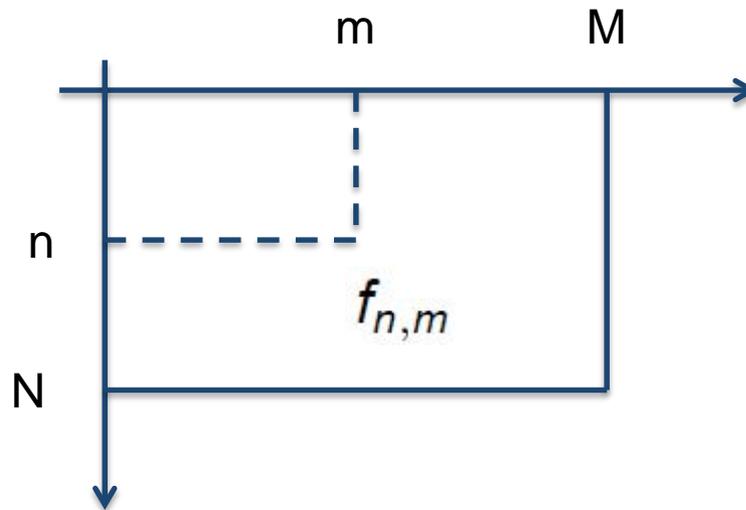
Joint teams:

JVT Joint Video Team (H.264/MPEG-4-AVC en 2003)

JCT-VC Joint Collaborative Team on Video Coding pour
développement de HEVC (High efficiency video codec)
(adopté Janv. 2013, ITU-T H.265 Avril 2013)

Qu'est ce qu'une image numérique ?

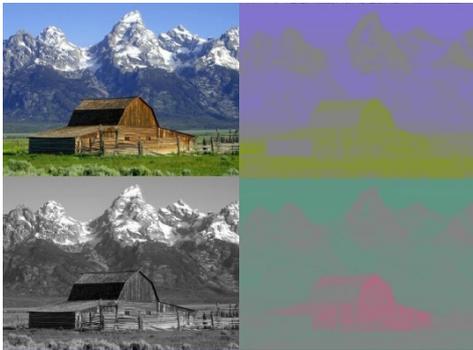
- Grille discrète de M (colonnes) x N (lignes) pixels
- Ordre de traitement canonique : balayage ligne par ligne unilatéral et de gauche à droite)



Décomposition d'une image couleur



R V B (rouge vert bleu)



YCbCr (appelé YUV)

Y luminance = $0,3R + 0,6V + 0,1B$

Cb Cr chrominance

$Cb \sim Y - B$

$Cr \sim Y - R$

compatible TV noir et blanc

oeil + sensible à Y que CbCr

Echantillonnage couleur

Le schéma est représenté comme un rapport entre trois termes :

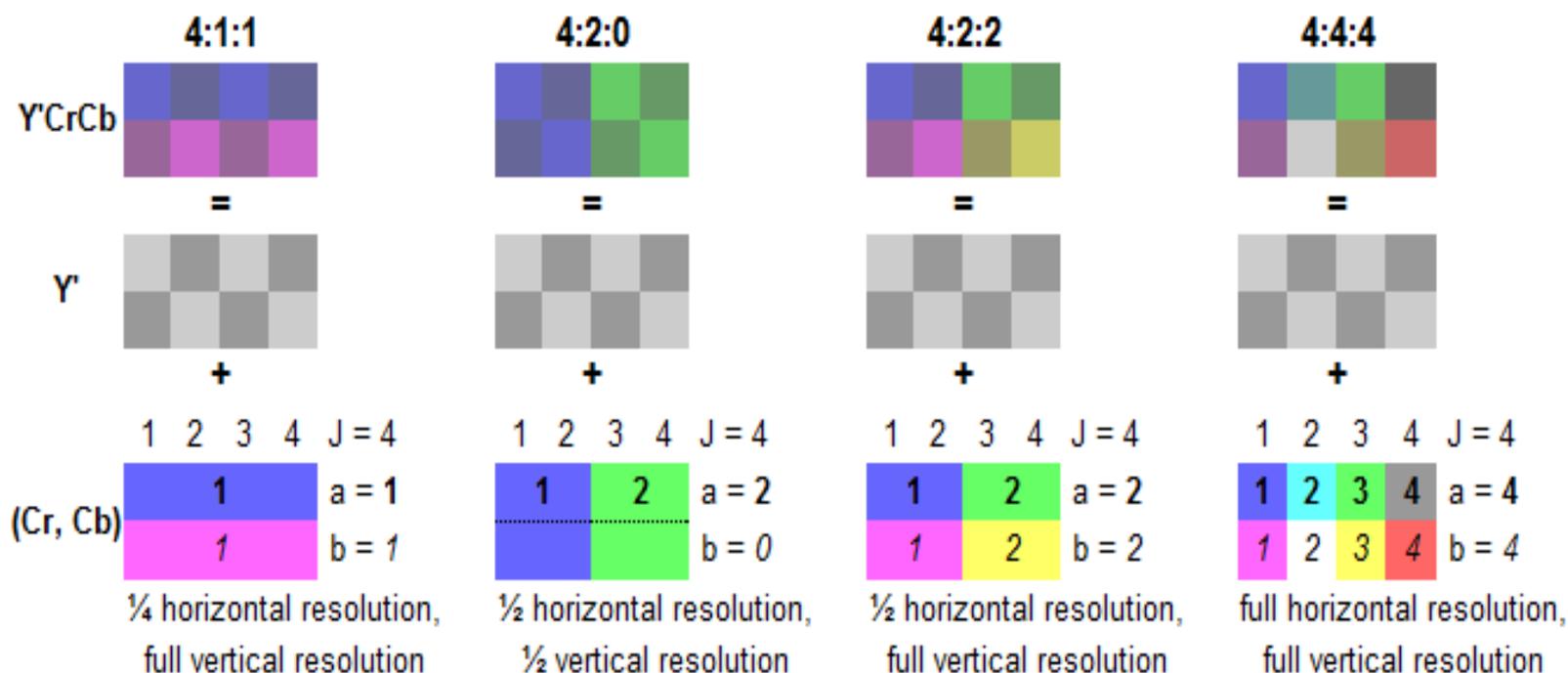
$J : a : b$

J Taille horizontale du bloc d'échantillonnage de référence,
normalement 4

a Nombre d'échantillons de chrominance dans la première ligne
de J pixels

b Nombre d'échantillons de chrominance *additionnels* dans la
deuxième ligne de J pixels

Echantillonnage couleur



Pourquoi compresser ?

i.e. transmettre même information avec moins de bits.

Soit une vidéo au format

HD 1080p = 1920×1080 à 50 Hz (images/sec)

	Y luminance	Cb Chrominance	Cr Chrominance
Nbre pixels	$1920 \times 1080 = 2.1 \text{ M}$	$960 \times 540 = .5 \text{ M}$	$960 \times 540 = .5 \text{ M}$
Nbre bits par pixel	8 bit	8 bit	8 bit
Débit	1.2 Gbps		

Quelques débits / capacité de stockage (qui évoluent au cours du temps)

- DVD 4.7 (1 face) à 17Go (2 faces) 31 à 113 s
- TNT 20 Mbps
- Ethernet/Fast Ethernet < 10 / 100 / 1000 Mbps
- DSL voie descendante 384 ... 2048 kbps
- Tél portable (débit pratique 2G /3G /4G) 9.05 / 384 kbps / 40 Mbps

Mesure de qualité

La qualité d'une image numérique est mesurée avec des **critères**,
fonctions de

$f_{n,m}$: image d'origine

$\tilde{f}_{n,m}$: image reconstruite après compression

Erreur quadratique moyenne (MSE) :

$$\mathcal{D} = \frac{1}{NM} \sum_{n=1}^N \sum_{m=1}^M (f_{n,m} - \tilde{f}_{n,m})^2$$

Rapport signal sur bruit crête :

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

SOMMAIRE

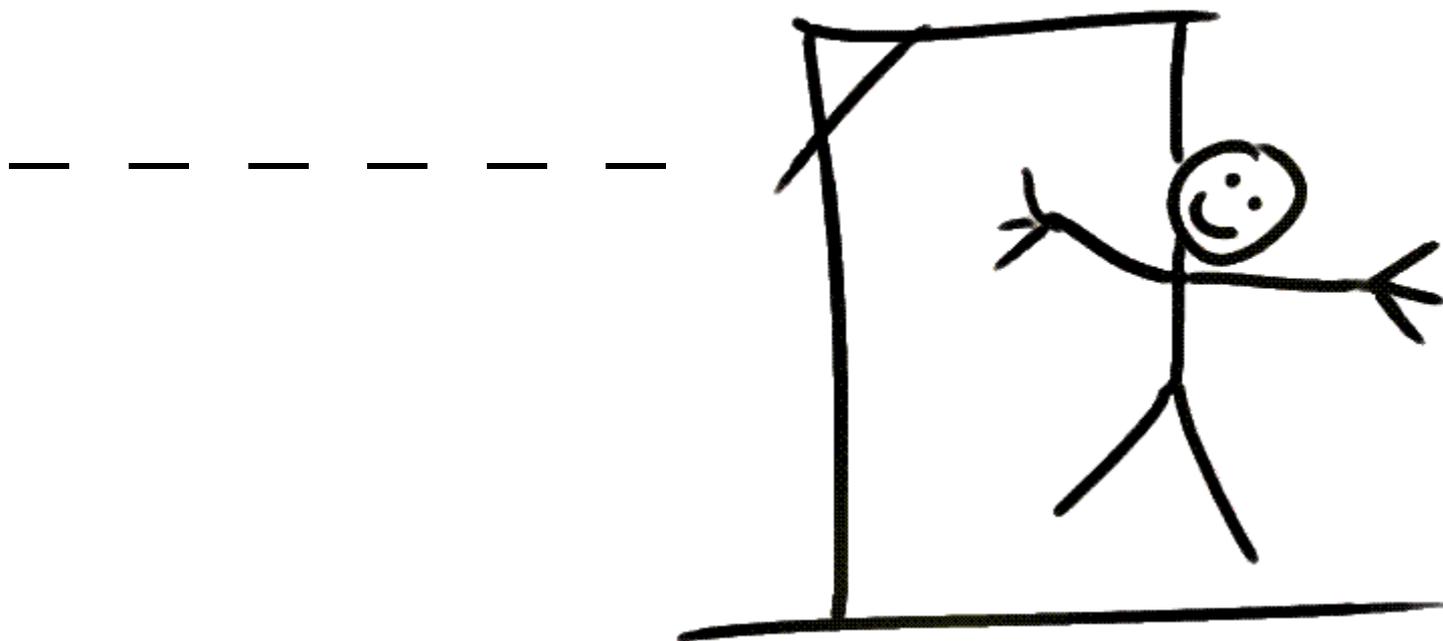
1. Pourquoi compresser une vidéo ?
- 2. Comment compresser ?**
3. Compression avec pertes
4. Compression par transformation
5. Compression par prédiction

2

Comment compressor ?

Le jeu du pendu

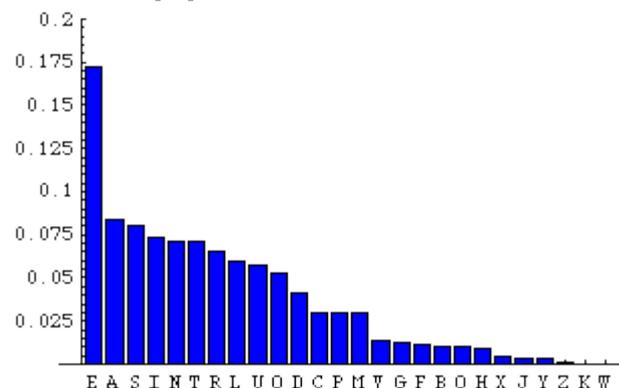
Objectif = jouer en dévoilant votre stratégie



2 idées pour gagner

1. Fréquence d'apparition des lettres

probabilité



2. Corrélation entre les lettres successives

dépendance

	_A	_B	_C	_D	_E	_F	_G	_H	_I	_J	_K	_L	_M	_N	_O	_P	_Q	_R	_S
A_	31	242	392	208	48	135	232	37	1255	32	7	663	350	1378	17	412	44	905	409
B_	158	2	1	2	130	1	2	0	132	4	10	181	1	1	146	1	3	187	29
C_	312	0	73	19	765	2	2	411	209	3	5	124	5	1	677	11	7	100	14
D_	427	1	8	24	2409	2	5	25	378	3	0	14	21	5	231	4	6	134	64
E_	616	176	917	998	782	258	209	67	179	96	8	1382	1056	2121	136	699	190	1514	3318

Analogie jeu du pendu - compression

- mot
- réponse à une question (oui/non).
- Objectif : proposer le moins de lettres par mot
- image
- 1 bit du train binaire qui représente image
- Objectif : proposer le moins de bits par image

Exploiter idée 1 : Utiliser la « non égalité » des probabilités

Soit une course avec les chevaux:

Nom	A natole	B arnabé	C hef	D arius
Probabilité de gagner	1/4	1/2	1/8	1/8

A gagne 1 fois sur 4, etc

Par exemple, la suite des vainqueurs peut être:

ABBCBBDABCBABBCBBDAABABC...

Question : coder par une suite de bits la suite des vainqueurs et de manière **efficace**...

Exploiter idée 1 : Utiliser la « non égalité » des probabilités

Soit une course avec les chevaux:

Nom	A natole	B arnabé	C hef	D arius
Probabilité de gagner	1/4	1/2	1/8	1/8
Mot de code	00	01	10	11

Coder la suite des vainqueurs peut être:

ABBCBBDABCBABBCBBDAABABC...
00 01 01 10 01 01 ... etc

Longueur moyenne 2 bits/course

Peut on mieux faire ?

Exploiter idée 2 : Utiliser la dépendance

Supposons que les courses de chevaux sont dépendantes

course 2 course 1	_A	_B	_C	_D		
A_	.25	.5	.125	.125		
B_	.25	.5	.125	.125		
C_	.25	.5	0	.25		
D_	.25	.5	.25	0		

Exploiter idée 2 : Utiliser la dépendance

	_A	_B	_C	_D	longueur moyenne	codage optimal
A_	.25	.5	.125	.125		
	10	0	110	111	1.75	1.75
B_	.25	.5	.125	.125		
	10	0	110	111	1.75	1.75
C_	.25	.5	0	.25		
	10	0		11	1.5	1.5
D_	.25	.5	.25	0		
	10	0	11		1.5	1.5

Exploiter idée 2 : Utiliser la dépendance

	_A	_B	_C	_D	longueur moyenne	codage optimal
A_	.25	.5	.125	.125		
	.25	10	0	110	111	1.75
B_	.25	.5	.125	.125		
	.5	10	0	110	111	1.75
C_	.25	.5	0	.25		
	.125	10	0		11	1.5
D_	.25	.5	.25	0		
	.125	10	0	11		1.5
					1.69	1.69

Complexité : taille du tableau à stocker croît *exponentiellement* avec le nbre de lettres traitées conjointement i.e. la **mémoire** de 4 à 16 = 4^2 mots de code

Notions de théorie de l'information

Processus aléatoire (source) $\{X_n\}$ Exemple : modèle des pixels

Longueur moyenne minimale (en bits) pour représenter un échantillon

- source **discrète i.i.d.** [Shannon]: **ENTROPIE**

$$H(X_1) = - \sum_x p(x) \log_2 p(x) \quad \text{bits/échantillon}$$

- source **discrète stationnaire ergodique** [Shannon-McMillan-Breiman]

$$\lim_{n \rightarrow +\infty} \frac{1}{n} H(X_1, \dots, X_n) \quad \text{bits/échantillon}$$

$$H(X_1, \dots, X_n) = - \sum_{x_1, \dots, x_n} p(x_1, \dots, x_n) \log_2 p(x_1, \dots, x_n)$$

Notions de théorie de l'information

Théorème : borne d'indépendance de l'entropie jointe

$$H(X_1, \dots, X_n) \leq H(X_1) + \dots + H(X_n)$$

↑
égalité ssi X_1, \dots, X_n indépendants

Interprétation

1. Exploiter la dépendance permet de réduire le nombre de bits. Mais nécessite de stocker une loi jointe (complexité croît exponentiellement avec n)
2. Des données indépendantes peuvent être codées séparément de manière optimale.

Notions de théorie de l'information

Exploisons idée : données dépendantes peuvent être codées séparément

- Transformer le vecteur X en un vecteur équivalent Y

$$H(X_1, \dots, X_n) = H(Y_1, \dots, Y_n) \lesssim H(Y_1) + \dots + H(Y_n)$$

Transformée inversible

Y indépendants

Transformer de manière à rendre les données indépendantes permet d'utiliser une compression par échantillon (codage entropique scalaire = codage avec la loi d'une donnée)

SOMMAIRE

1. Pourquoi compresser une vidéo ?
2. Comment compresser ?
3. **Compression avec pertes**
4. Compression par transformation
5. Compression par prédiction

3

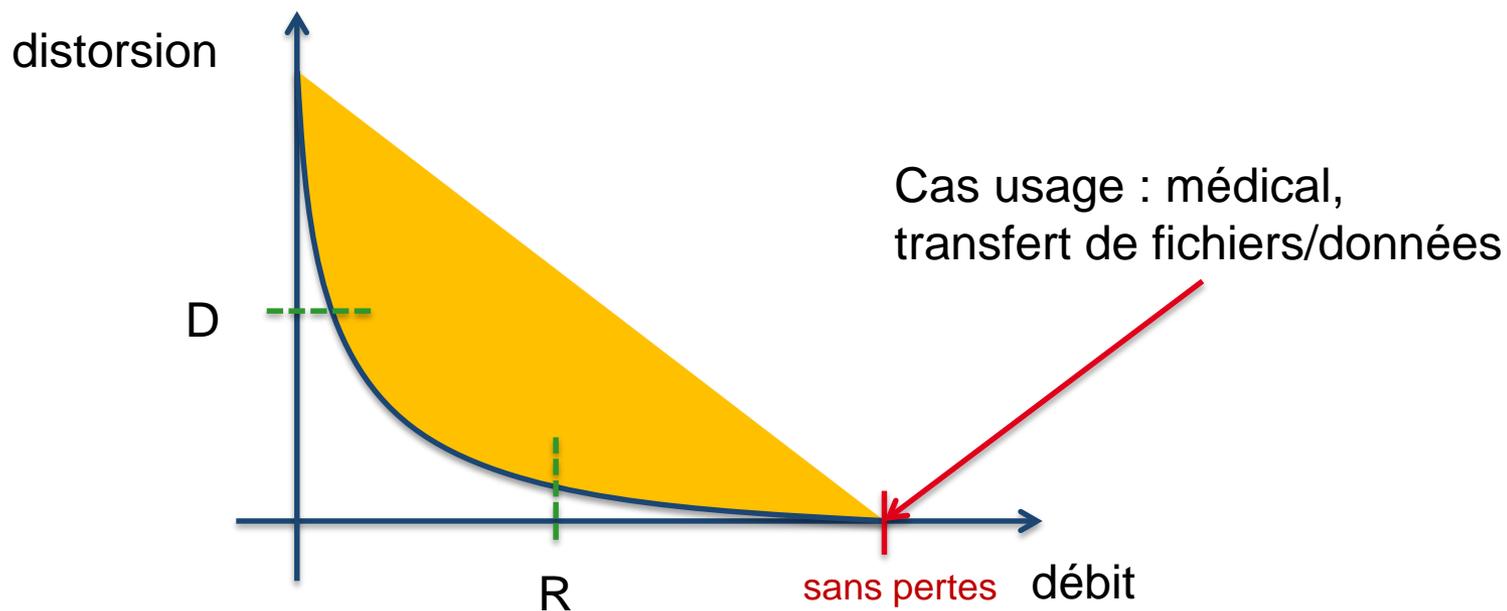
Compression avec pertes

Compression en autorisant des « pertes »

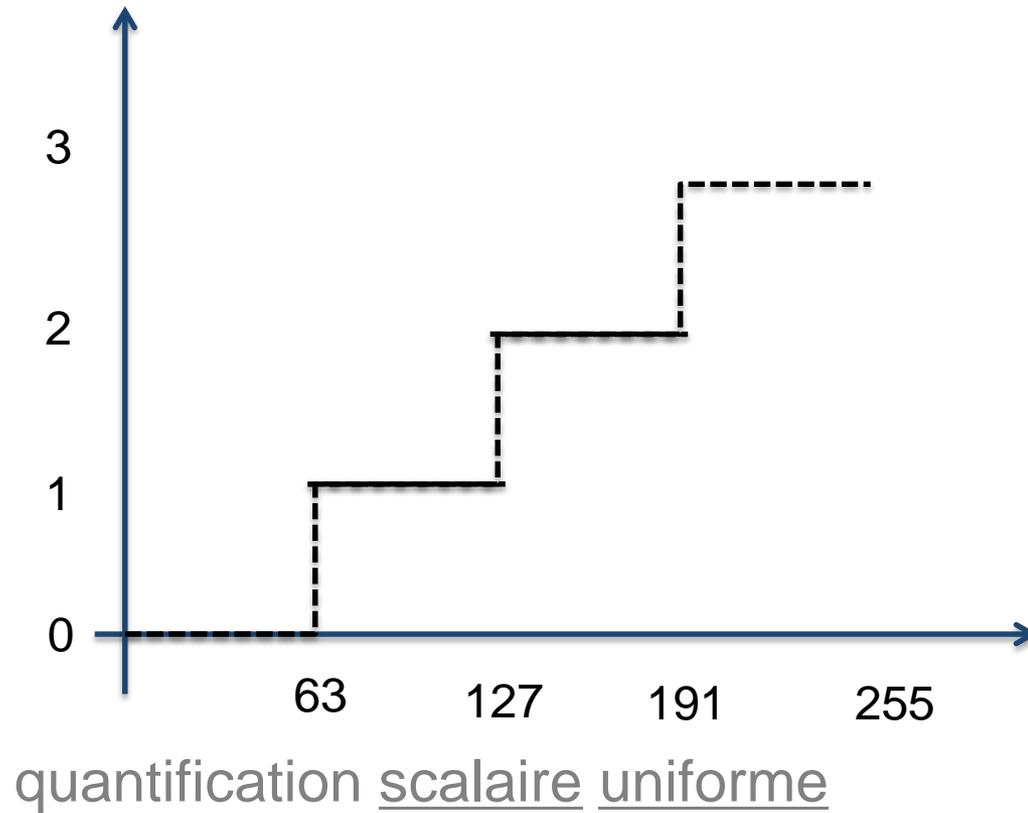
Compression avec contrainte de débit

débit = nbre bits / pixel (ou lettre)

distorsion = distance entre l'image originale et celle encodée

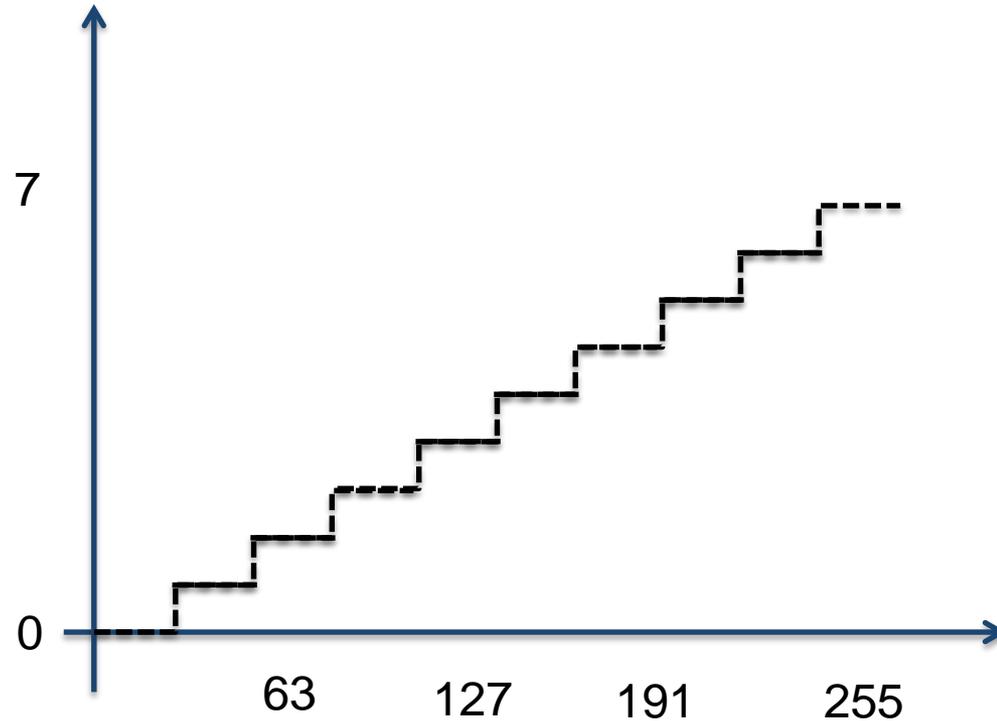


Mise en œuvre de la compression avec pertes



8 bits → 2 bits

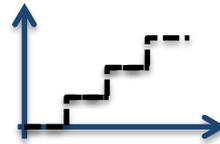
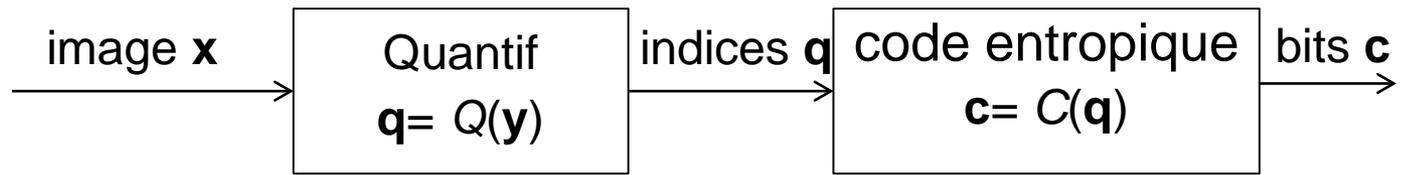
Mise en œuvre de la compression avec pertes



quantification scalaire uniforme

8 bits \rightarrow 3 bits

Mise en œuvre efficace de la compression avec pertes



A	B	C	D
1/4	1/2	1/8	1/8
10	0	110	111

quantification scalaire uniforme + codage entropique

→ dégradation de **1.3 dB** par rapport à optimal

SOMMAIRE

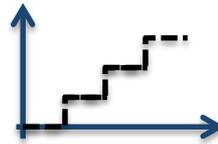
1. Pourquoi compresser une vidéo ?
2. Comment compresser ?
3. Compression avec pertes
4. **Compression par transformation**
5. Compression par prédiction

4

Données dépendantes

la transformation

Où traiter la dépendance ?

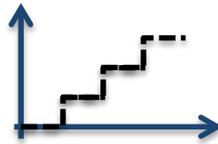
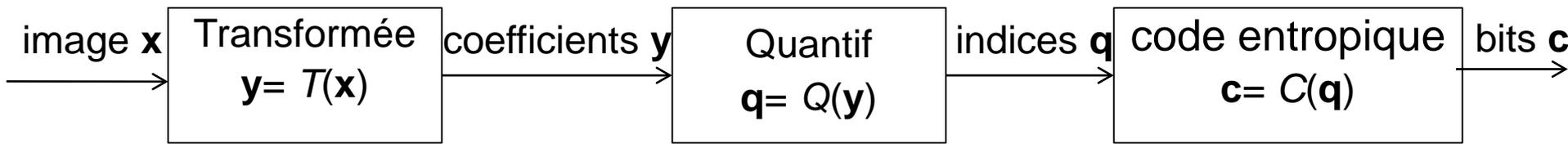


A	B	C	D
1/4	1/2	1/8	1/8
10	0	110	111

Schéma ci-dessus quasi-optimal si les pixels indépendants

.....

Ou traiter la dépendance ?



A	B	C	D
1/4	1/2	1/8	1/8
10	0	110	111

Schéma ci-dessus quasi-optimal si les pixels indépendants
... transformer les pixels pour les rendre indépendants...

Exploitation de la dépendance des données

		a	b	
		c	d	

$(a,b,c,d) \rightarrow (m, b-m, c-m, d-m)$: transformation
inversible

Dépendance =

Pixels voisins ont des valeurs proches

$$m = (a+b+c+d)/4$$

		m	b-m	
		c-m	d-m	

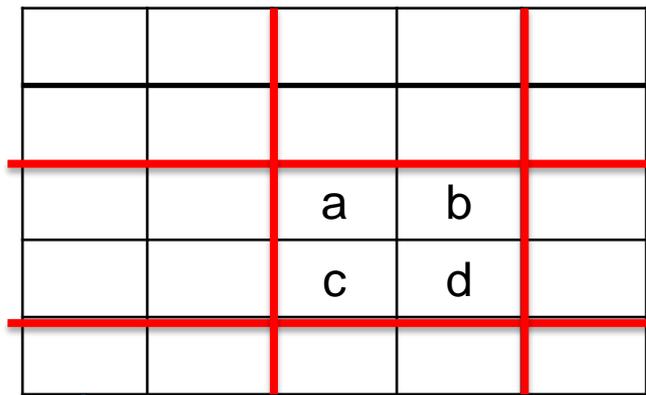
Différence à la moyenne :

- de faible amplitude
- casser la dépendance dans nouveau bloc ?

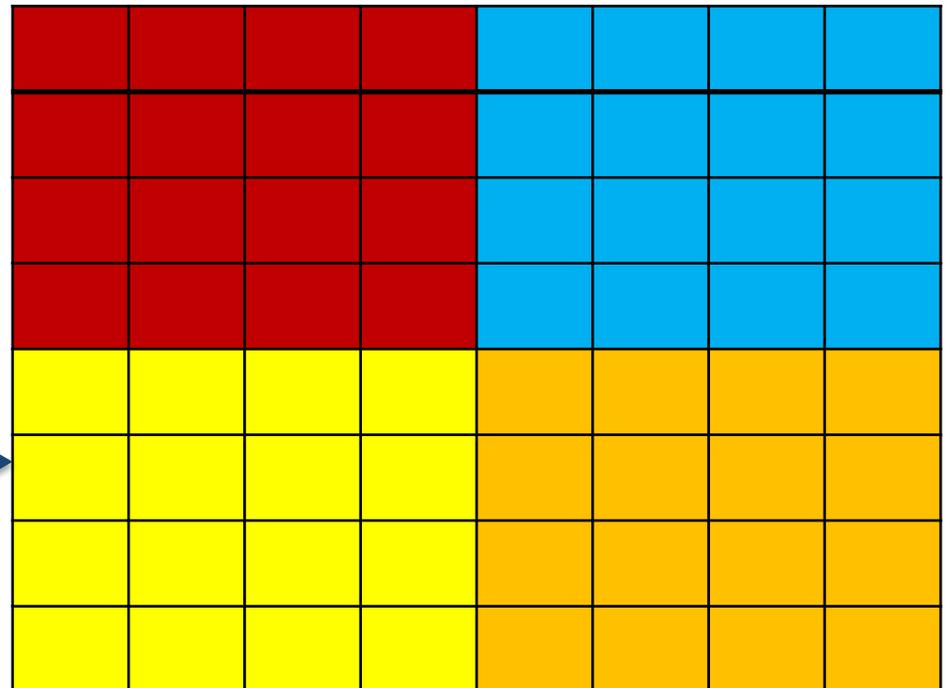
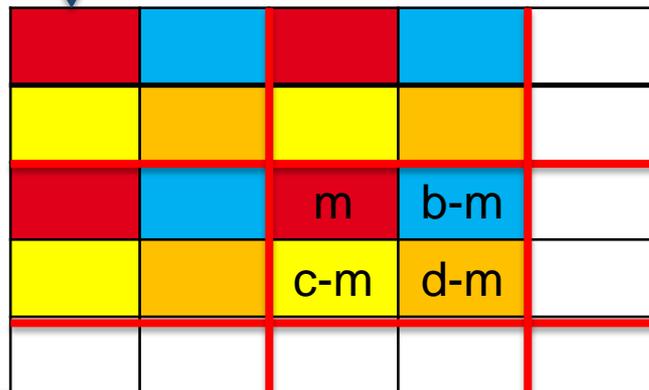
Exploitation de la dépendance des données

Cette transformation est un cas particulier :

- DCT transformée en cosinus discret (MPEG 1, 2, 4, HEVC)
- ondelettes (JPEG-2000, Dirac)



$$m = (a + b + c + d) / 4$$





Original Lena (512 x 512 Pixels, 8-Bit Y/ pixel, 257Ko)

Gris = 0 car m-d codé en m-d+128



Reste de dépendance

Gris = 0



Gris = 0



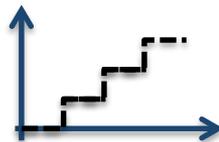
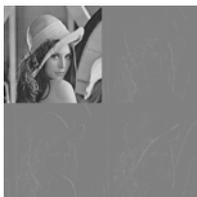
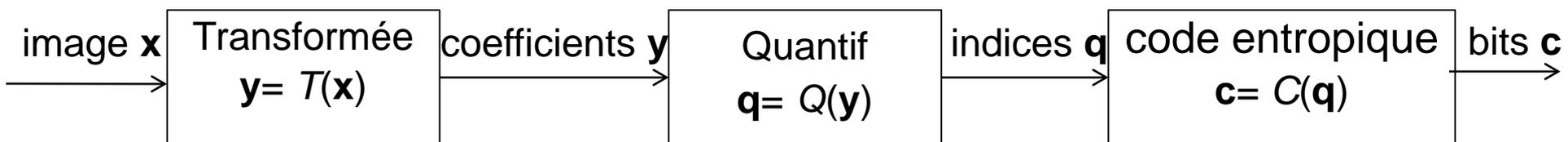
exemple d'ondelettes

Transformée utilisée dans JPEG 2000



Moins de corrélation que précédemment entre les 4 blocs...
mais il en reste encore

JPEG2000



A	B	C	D
1/4	1/2	1/8	1/8
10	0	110	111

Résultats de compression JPEG2000



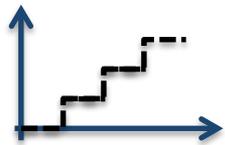
Original Carol (512 x 512 Pixels, 24-Bit RGB/pixel, Taille 786ko)

Résultats de compression JPEG2000



75:1, 10.6 ko

Résultats de compression JPEG2000



150:1, 5.3 ko

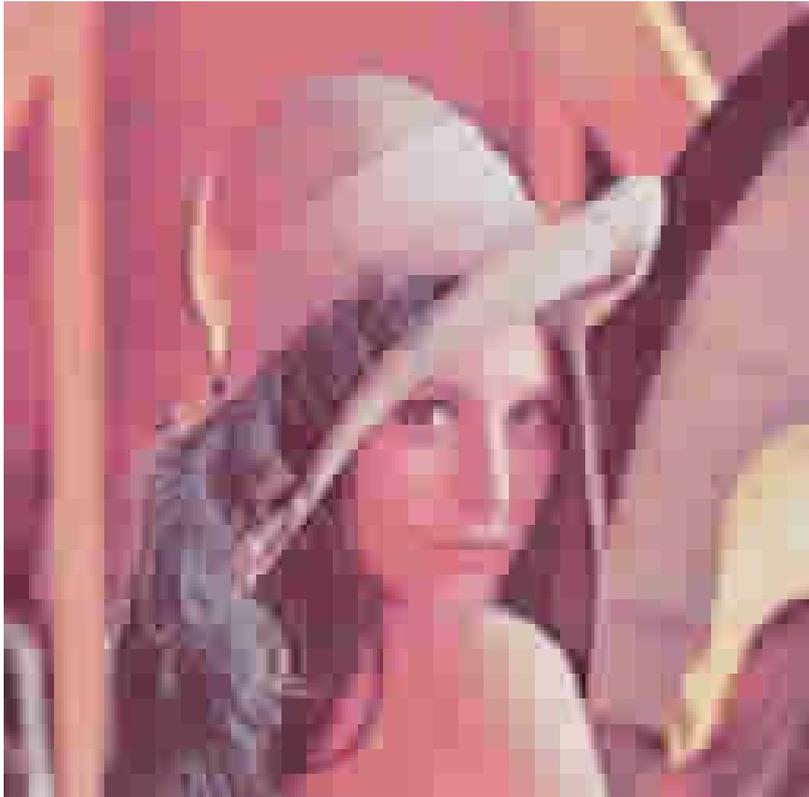
Résultats de compression JPEG2000



300:1, 2.6 ko



Comparaison JPEG (1992) vs. JPEG2000



Lena, 256x256 RGB
Baseline JPEG: 4572 octets



Lena, 256x256 RGB
JPEG-2000: 4572 octets

SOMMAIRE

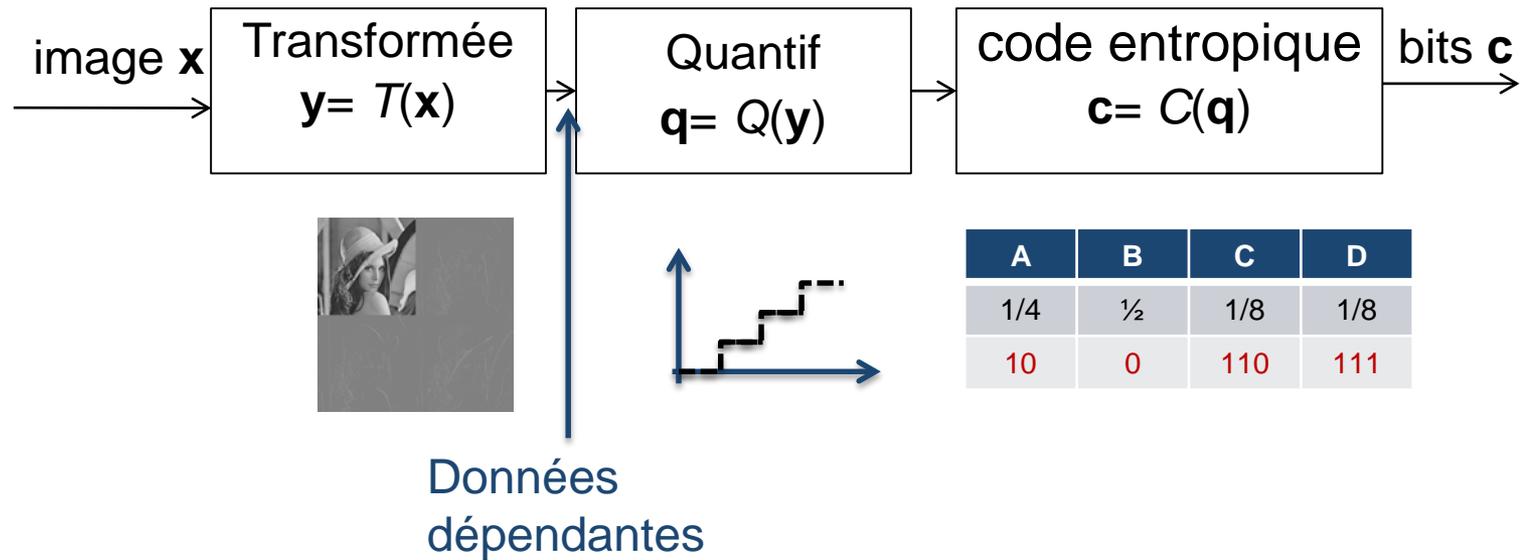
1. Pourquoi compresser une vidéo ?
2. Comment compresser ?
3. Compression avec pertes
4. Compression par transformation
5. **Compression par prédiction**

5

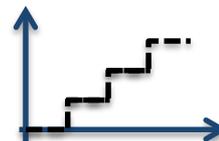
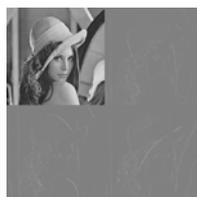
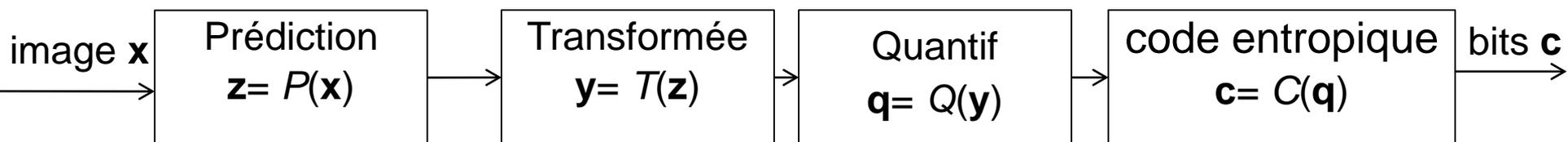
Données dépendantes

la prédiction

Il reste de la dépendance



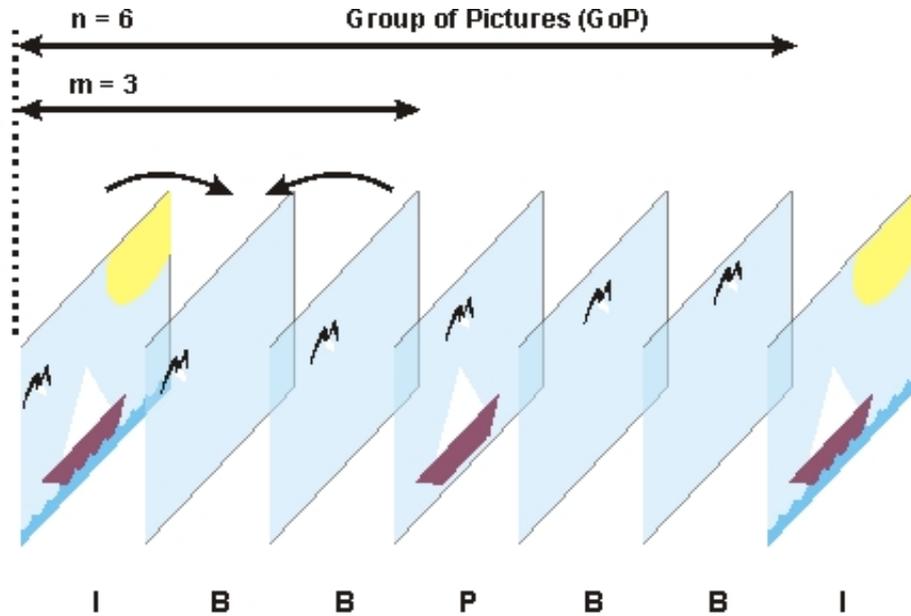
Il reste de la dépendance



A	B	C	D
1/4	1/2	1/8	1/8
10	0	110	111

Prédiction: pour mieux supprimer la redondance

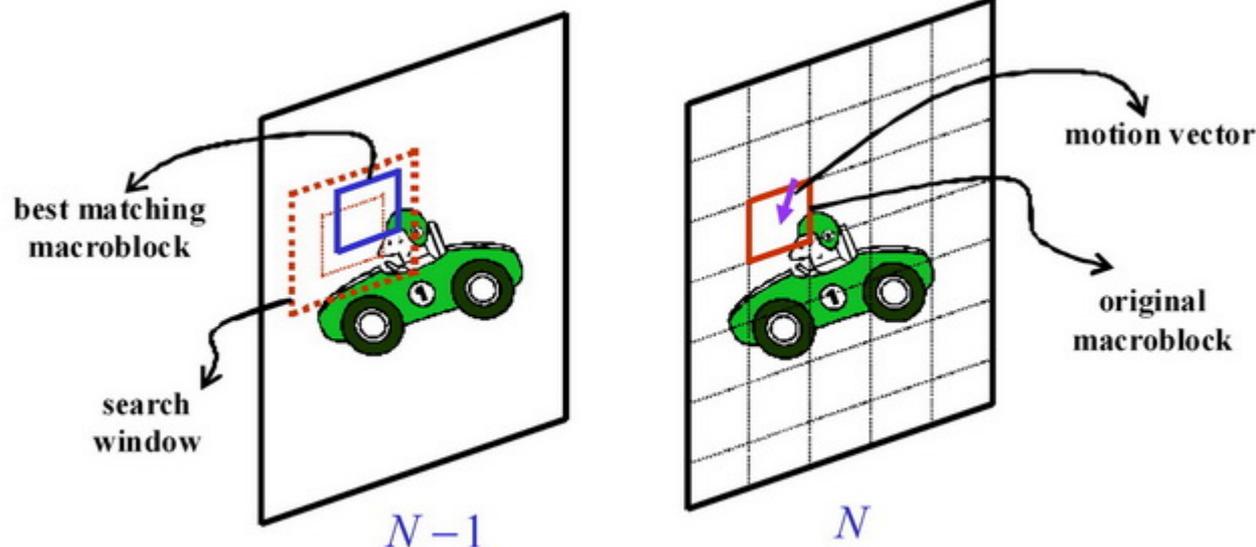
Prédiction... entre images



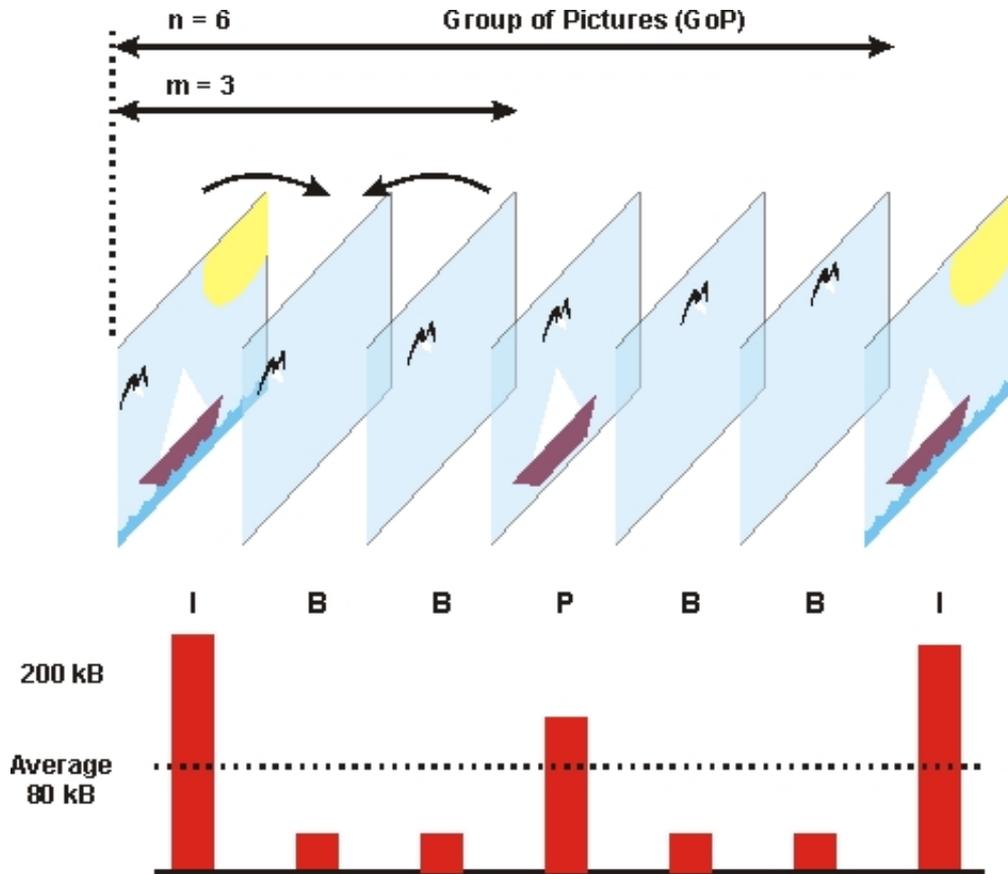
Intra: image codée indépendamment des autres (synchro)

Inter: (P/B) image codée en fonction d'autres images

Utilisé en MPEG-2, -4, HEVC



Prédiction... entre images



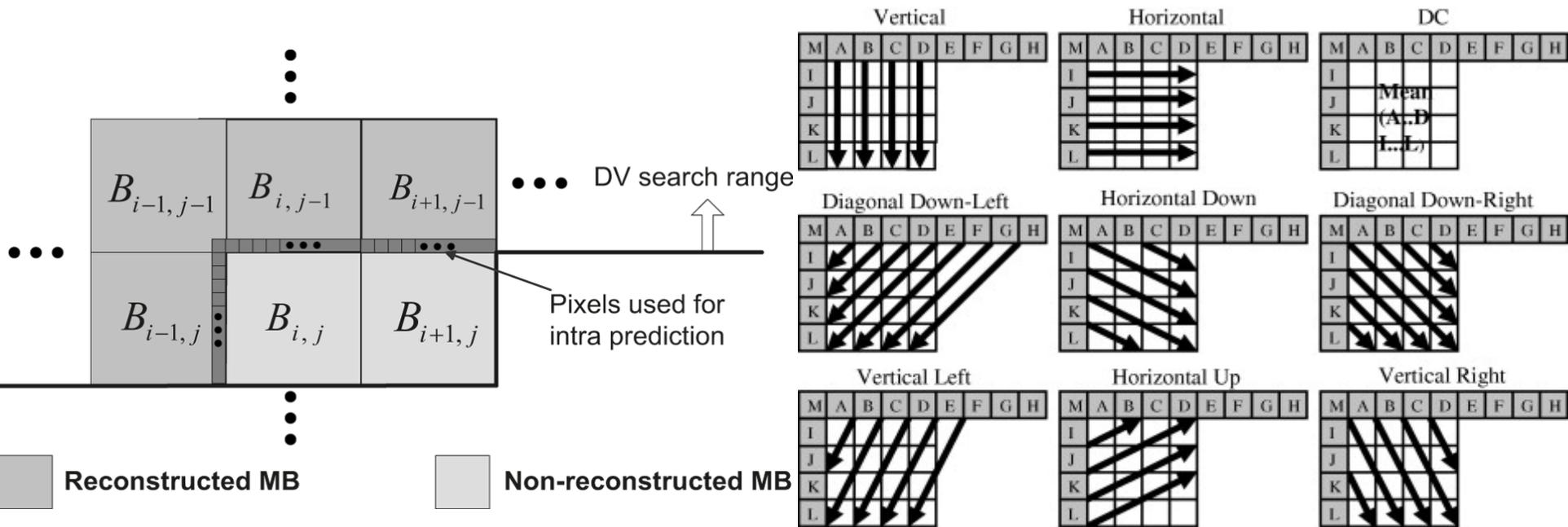
Intra: image codée indépendamment des autres (synchro)

Inter: (P/B) image codée en fonction d'autres images

Utilisé en MPEG-2, -4, HEVC

Très efficace: débit(inter) \ll débit (intra)

Prédiction Intra (MPEG-4, HEVC)



Si prédiction fiable, différence entre original et prédiction (résidu) :

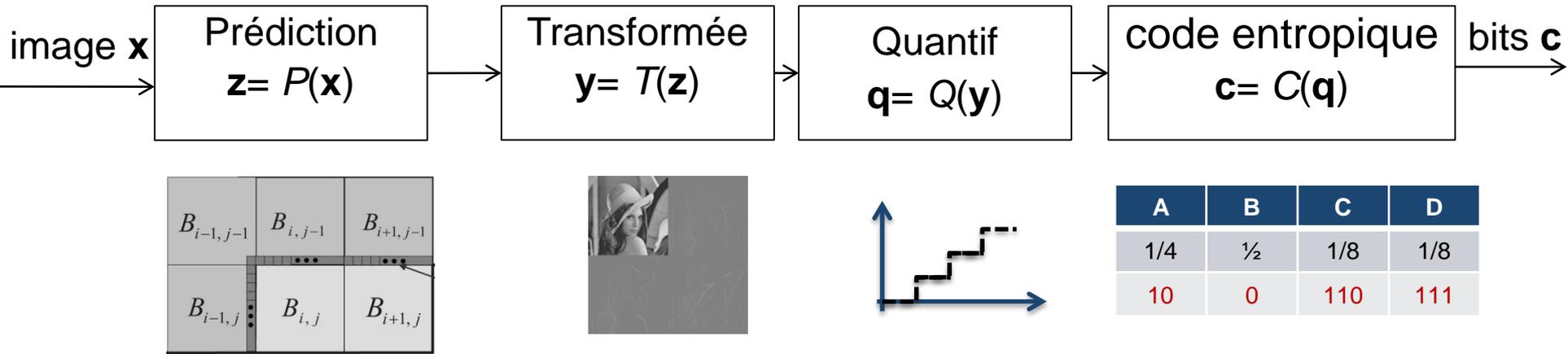
- échantillons indépendants [théorie de l'estimation]

6

Conclusion

les messages à ramener à la maison

Take-home-message



Transformation et prédiction : exploite dépendance des pixels

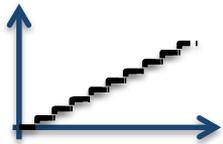
Quantification : introduit pertes

Codage entropique : exploite non égalité des probabilités des pixels

Quelques résultats en vidéo de HEVC

Vidéo BasketballPass 416x240 pixels 50Hz (8 + (8+8)/4 bits) 60 Mbps

Pas de quantification	débit	Taux compression
non compressé	60 Mbps	
22	1 Mbps	60:1
32	340 kbps	176:1
51	30 kbps	2000:1



original

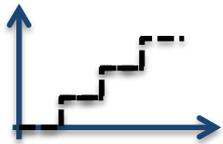


60:1

Quelques résultats en vidéo de HEVC

Vidéo BasketballPass 416x240 pixels 50Hz (8 + (8+8)/4 bits) 60 Mbps

Pas de quantification	débit	Taux compression
non compressé	60 Mbps	
22	1 Mbps	60:1
32	340 kbps	176:1
51	30 kbps	2000:1



original



176:1

Quelques résultats en vidéo de HEVC

Vidéo BasketballPass 416x240 pixels 50Hz (8 + (8+8)/4 bits) 60 Mbps

Pas de quantification	débit	Taux compression
non compressé	60 Mbps	
22	1 Mbps	60:1
32	340 kbps	176:1
51	30 kbps	2000:1

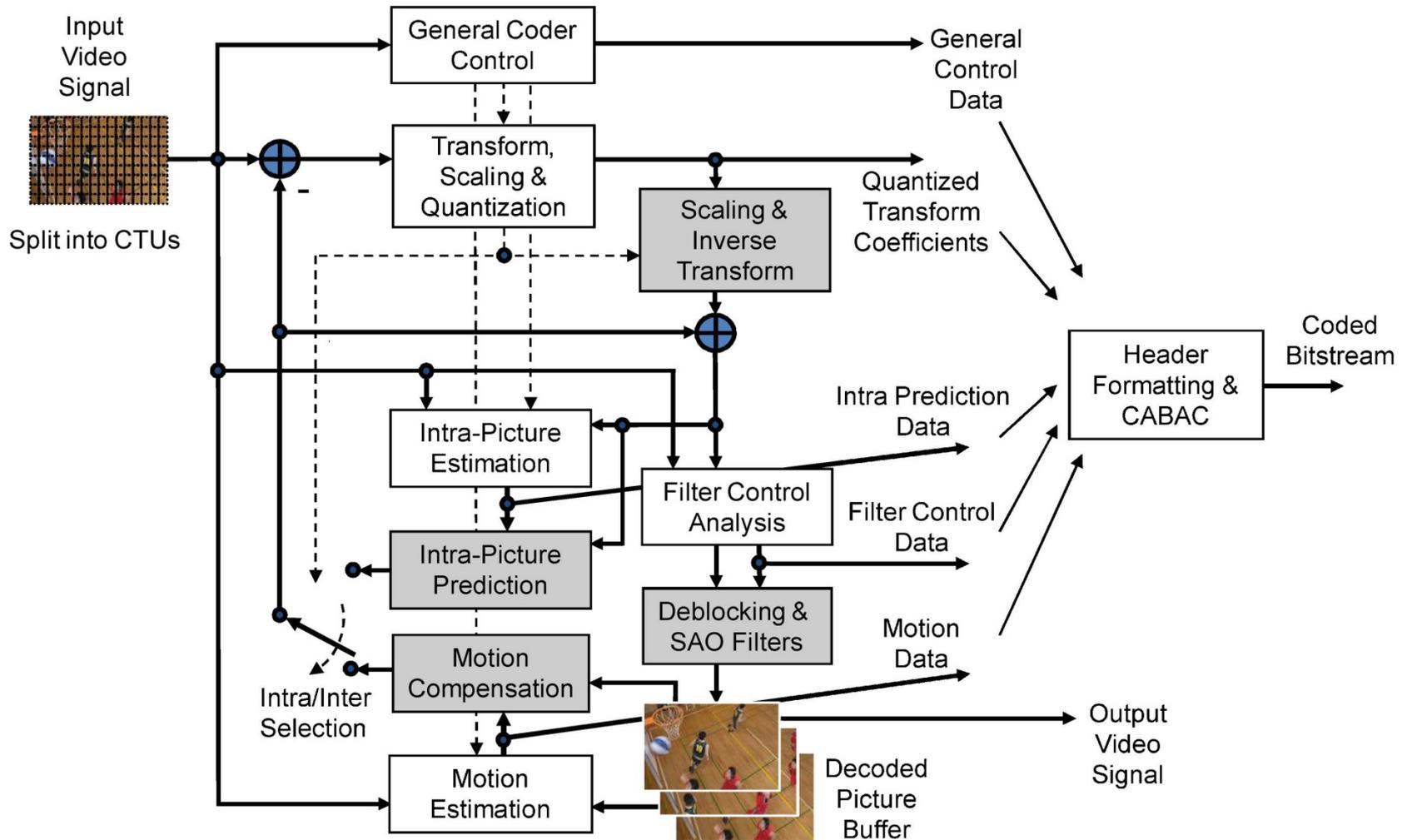


original



2000:1

Le schéma de codage HEVC



Les éléments grisés correspondent au décodeur (Overview of HEVC Standard, TCSVT Dec 2012)

ATELIER

Mise en œuvre de la première étape d'un codeur vidéo :
La prédiction compensée en mouvement.

Pour en savoir plus...

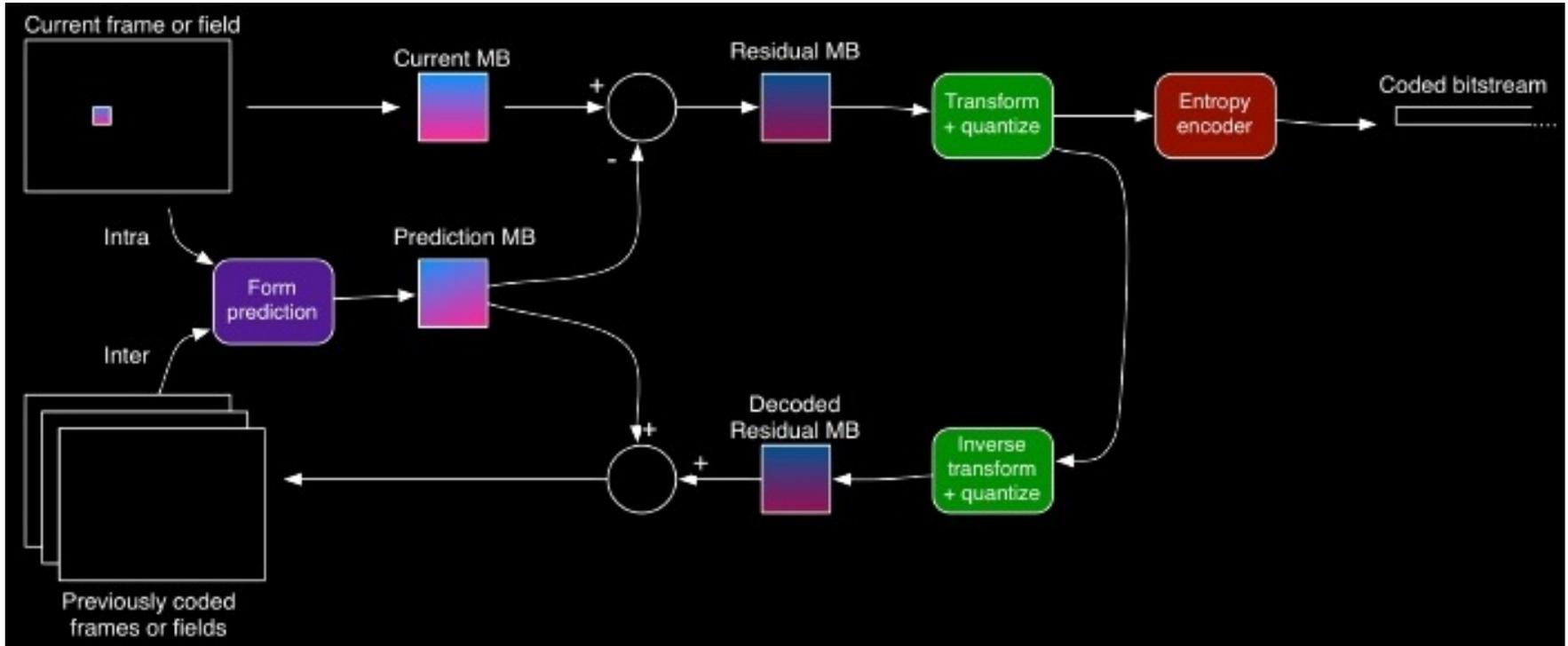
1. Thomas Wiegand and Heiko Schwarz:
Source Coding: Part I of Fundamentals of Source and Video Coding,
Foundations and Trends in Signal Processing, vol. 4, no. 1-2, pp. 1-222, January 2011.
<http://iphome.hhi.de/wiegand/pubs.htm>
2. Ian Richardson
The Advanced H 264 video compression standard, Wiley 2010.
3. Vivienne Sze, Madhukar Budagavi, Gary J. Sullivan
High Efficiency Video Coding (HEVC), Springer 2014.

merci

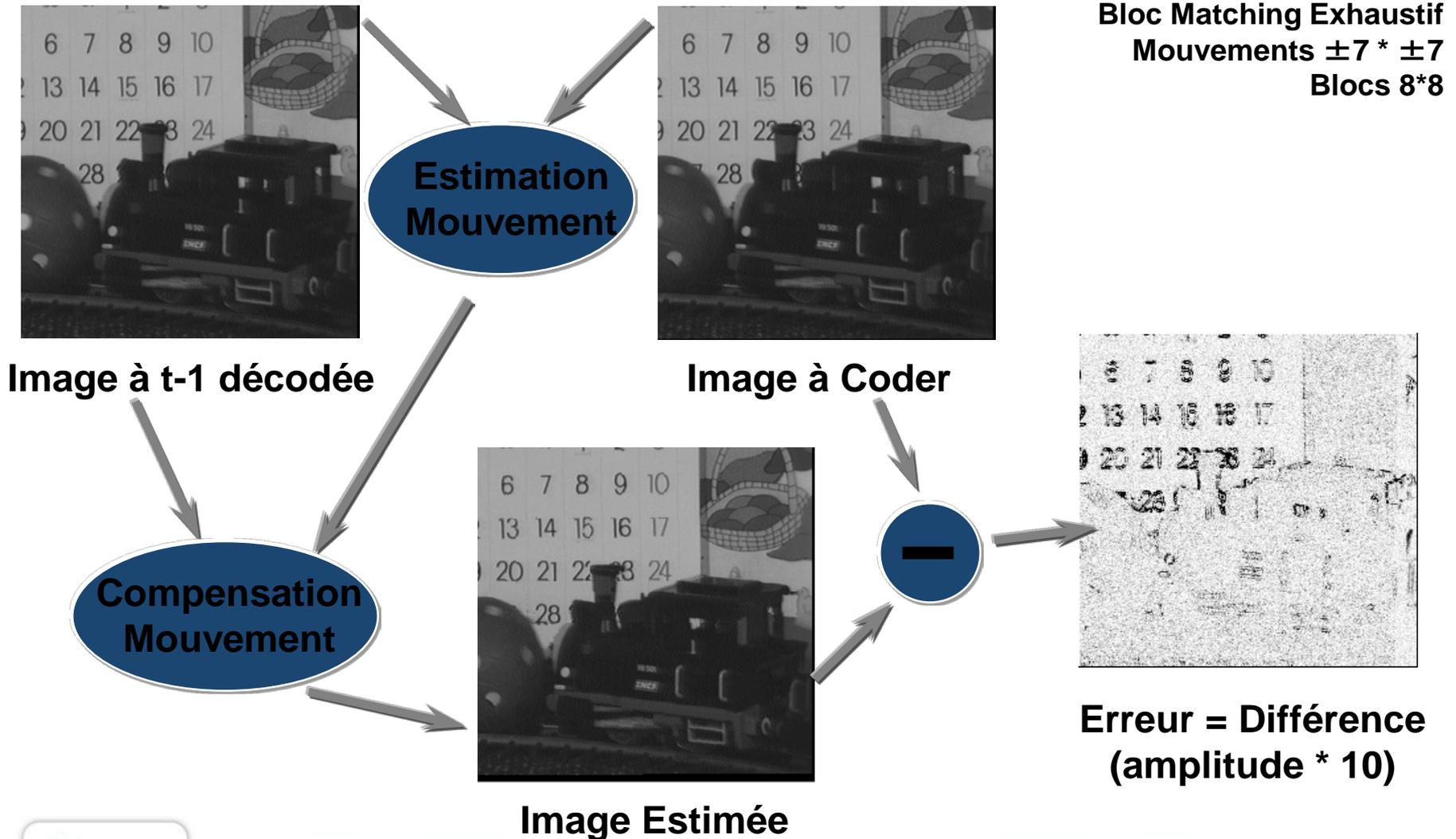


www.inria.fr

Codeur vidéo



Prédiction temporelle compensée en mouvement



Prédiction temporelle compensée en mouvement

