



Compression d'images : pourquoi, comment ?

Aline ROUMY

Chercheur, INRIA Rennes

SOMMAIRE

- 1.** Pourquoi compresser une vidéo ?
- 2.** Comment compresser ?
- 3.** Compression avec pertes
- 4.** Compression par transformation
- 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 (adopté Janv. 2013,
ITU-T H.265 Avril 2013)

Décomposition d'une image couleur



R V B (rouge vert bleu)

YCbCr

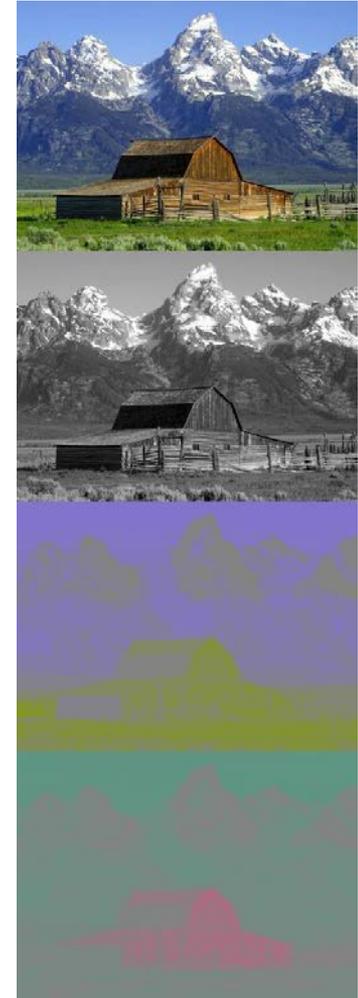
Y luminance = $0,3R + 0,6V + 0,1B$

Cb Cr chrominance

Cb $\sim Y - B$

Cr $\sim Y - R$

compatible TV noir et blanc
oeil + sensible à Y que CbCr



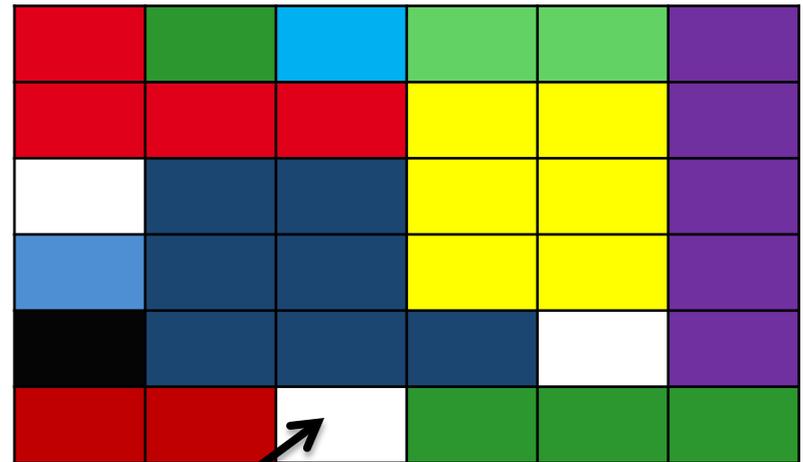
Qu'est ce qu'une image numérique ?

image analogique



de La Tour - *Le nouveau né* - Rennes

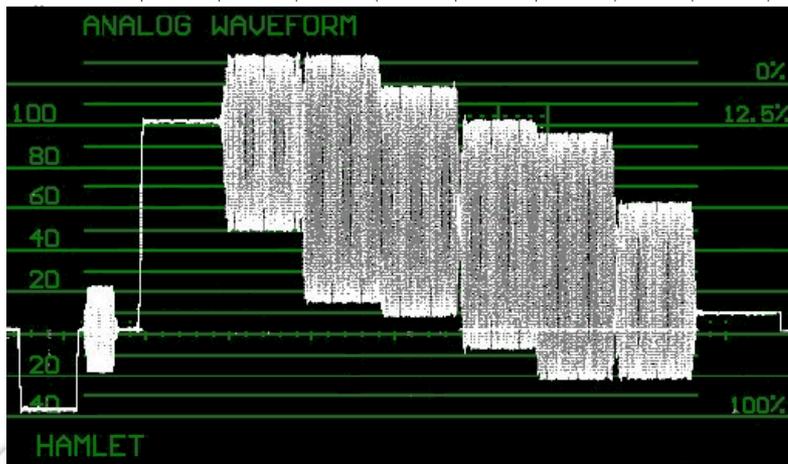
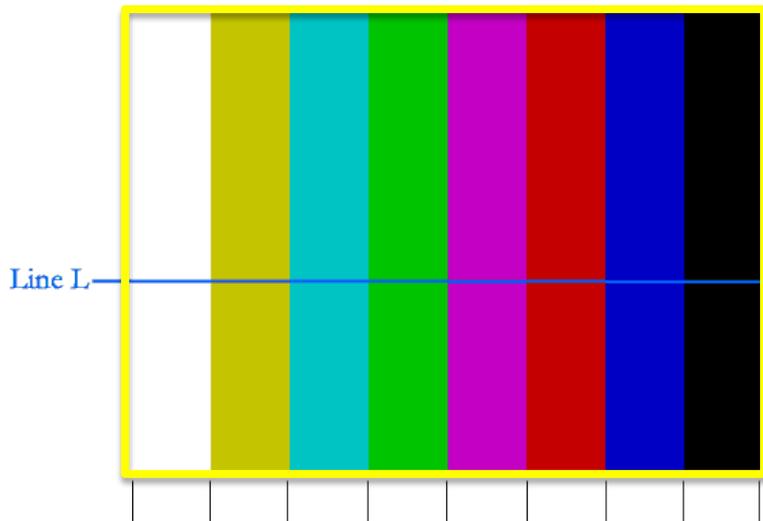
image numérique



pixel

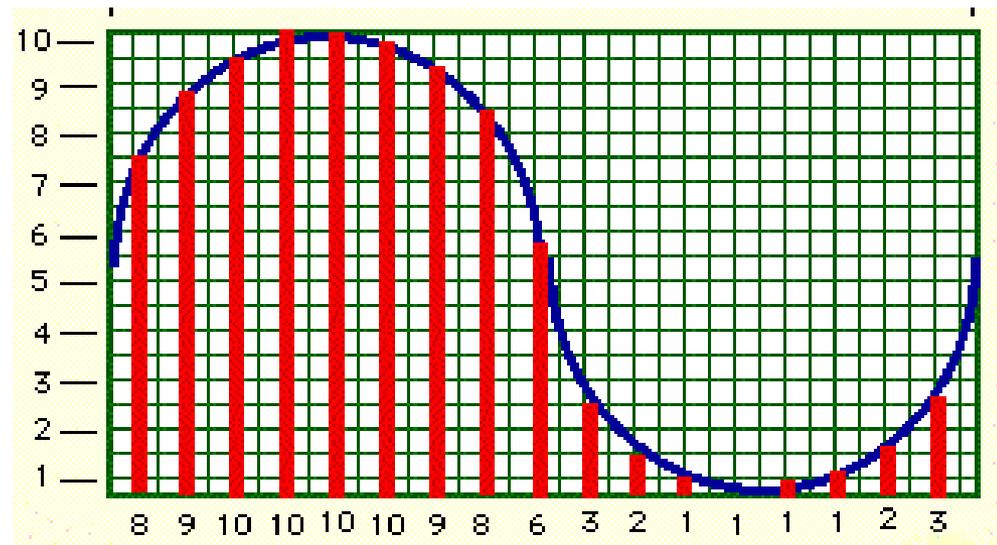
Conversion analogique - numérique

TV analogique (SECAM)



conversion analogique numérique

échantillonner + quantifier



ITU-R Rec. (Recommendation) 601

Luminance : 13.5 M échantillons / seconde

Chrominance : 6.75 M échantillons / seconde

8 bits par échantillon

Pourquoi compresser ?

i.e. transmettre même information avec moins de bits

	Y Luminance	Cb Chrominance	Cr Chrominance
Nbre échantillons par sec	13.5 M	6.75 M	6.75 M
Nbre bits par échantillons	8 bit	8 bit	8 bit
Débit	216 Mbps		

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

- DVD 4.7 (1 face) à 17Go (2 faces) 18 à 78 s
- TNT 20 Mbps
- Ethernet/Fast Ethernet < 10 / 100 / 1000 Mbps
- DSL voie descendante 384 ... 2048 kbps
- Tél portable (débit pratique 2G /3G) 9.05 ... 384 kbps

SOMMAIRE

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

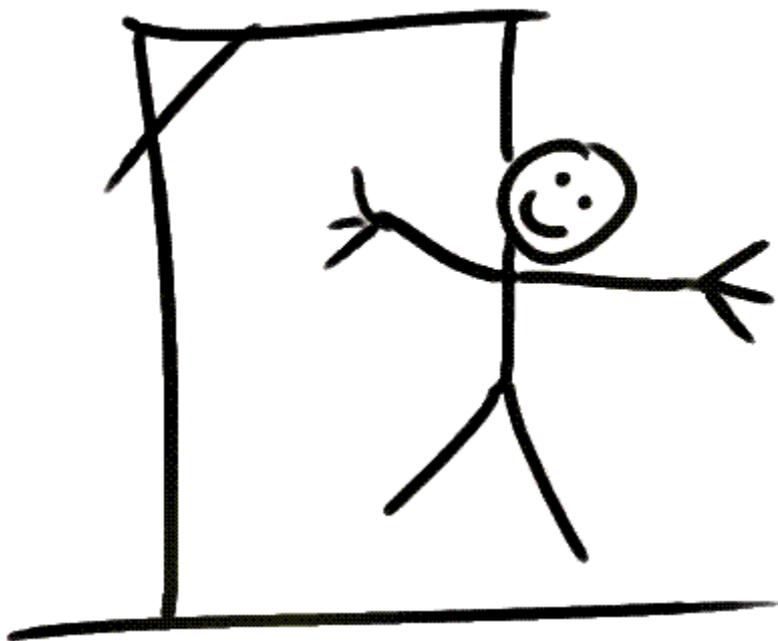
2

Comment compressor ?

Le jeu du pendu

Objectif = jouer en dévoilant votre stratégie

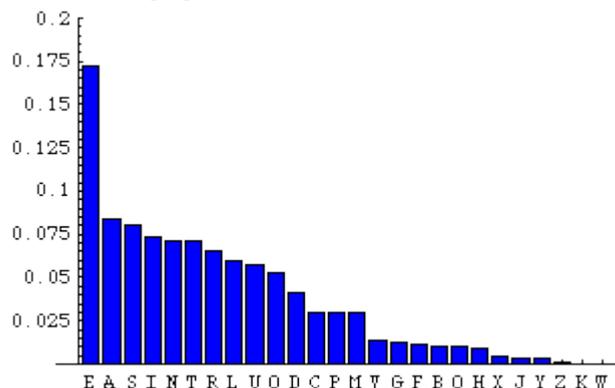
— — — — —



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
- image
- joueur (qui cherche mot : lettre proposée reçoit oui/non) .
- codeur (celui qui compresse : transforme l'image en suite de bits)
- décodeur (qui cherche image : à partir des bits fournis par codeur)
- Objectif : proposer le moins de lettres possibles par mot
- Objectif : proposer le moins de bits possibles 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 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	10	0	110	111

Coder la suite des vainqueurs peut être:

ABBCBBDABCBABBCBBDAABABC...

Longueur codage à longueur fixe 2 bits/course

Longueur moyenne $2 \cdot \frac{1}{4} + 1 \cdot \frac{1}{2} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8}$ 1,75 bits/course

Longueur optimale [théorie de l'information⁴⁸] 1,75 bits/course

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

THM (take-home-message)

On peut compresser des données numériques (texte, **audio**, **image**) en utilisant

1. Fréquence d'apparition des lettres **probabilité**
 - Les lettres les plus fréquentes sont codées avec moins de bits
2. Corrélacion entre les lettres **dépendance**
 - Coder une lettre en tenant compte du/des précédentes lettres

Un codeur qui utilise ces 2 idées est un **codeur entropique**

Complexité :

Il faut stocker un tableau dont la taille croît **exponentiellement** avec

- la mémoire (2.) (durée du passé)

Donc dépendance/mémoire traitée autrement...

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

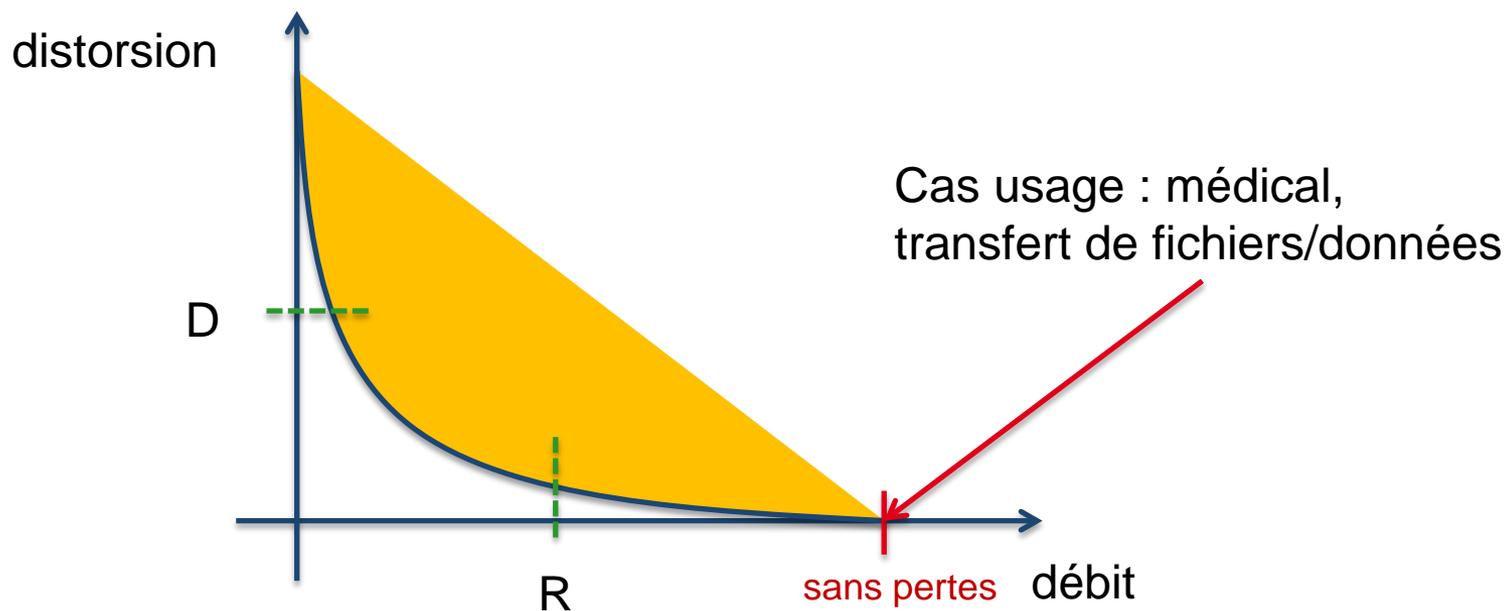
Compression en autorisant des « pertes »

Supposons les lettres/pixels indépendants

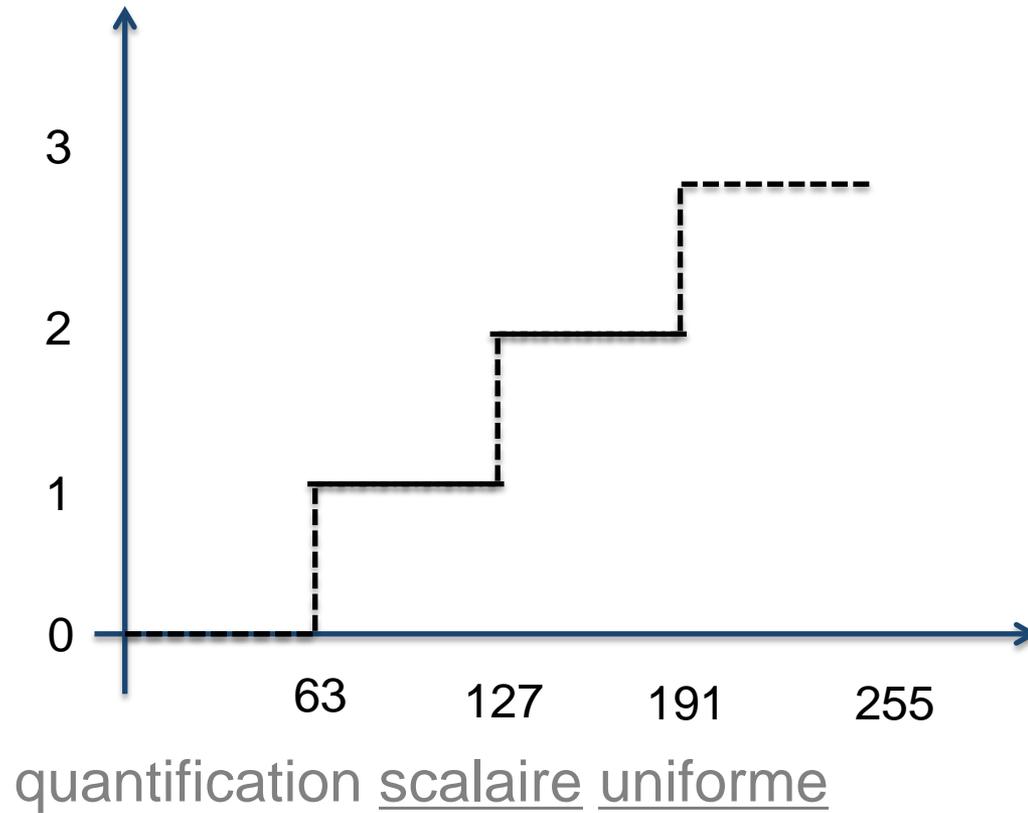
distorsion, pertes, dégradation =

distance entre l'image originale et celle encodée

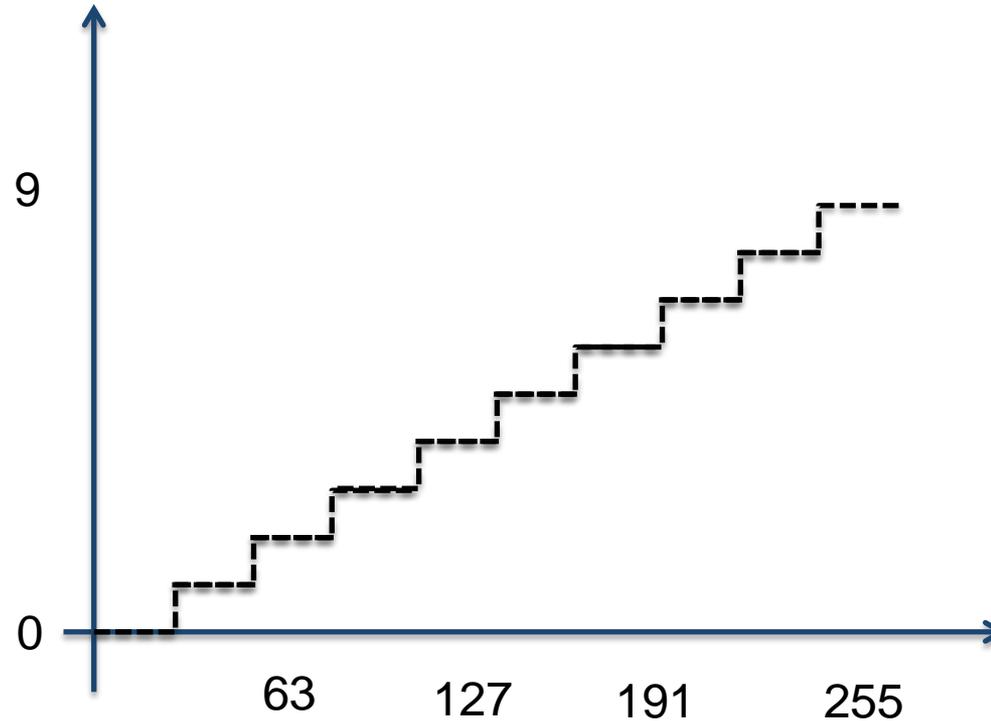
débit = nbre bits / pixel (ou lettre)



Mise en œuvre de la compression avec pertes

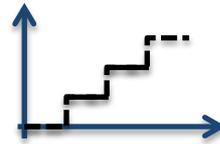
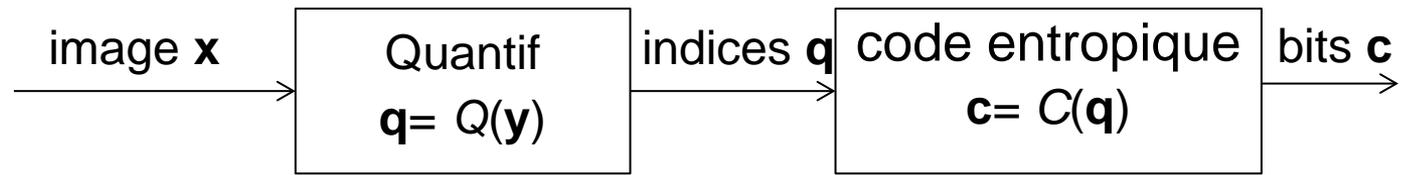


Mise en œuvre de la compression avec pertes



quantification scalaire uniforme

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

→ optimal + **0.25** bits / pixel ... si pixels *indépendants*

SOMMAIRE

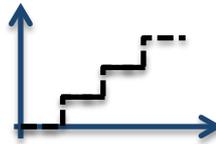
1. Pourquoi compresser une vidéo ?
2. Comment compresser ?
3. Compression avec pertes
4. **Compression par transformation**
5. Compression par prédiction

4

Données dépendantes

la transformation

Où traiter la dépendance ?

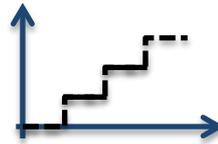
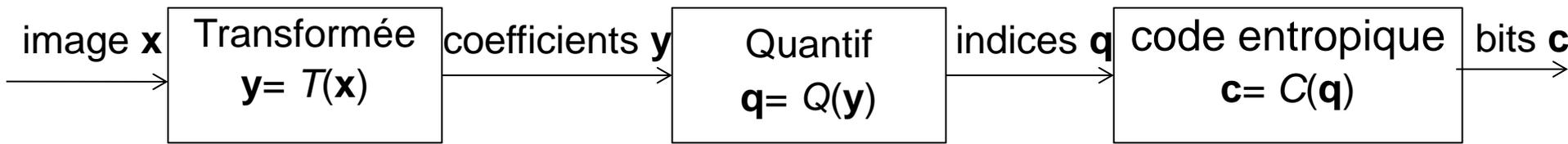


A	B	C	D
1/4	1/2	1/8	1/8
10	0	110	111

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	

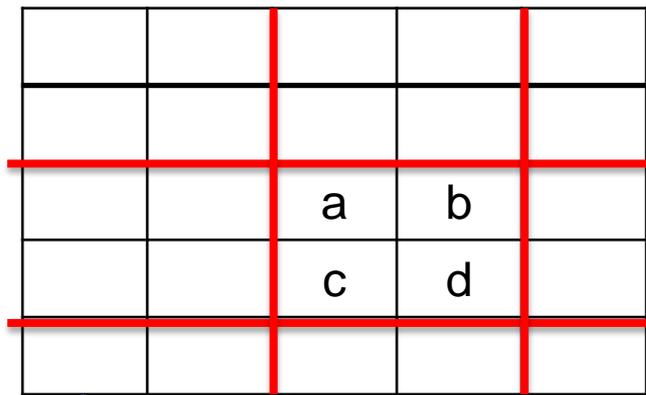
Différence à la moyenne :

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

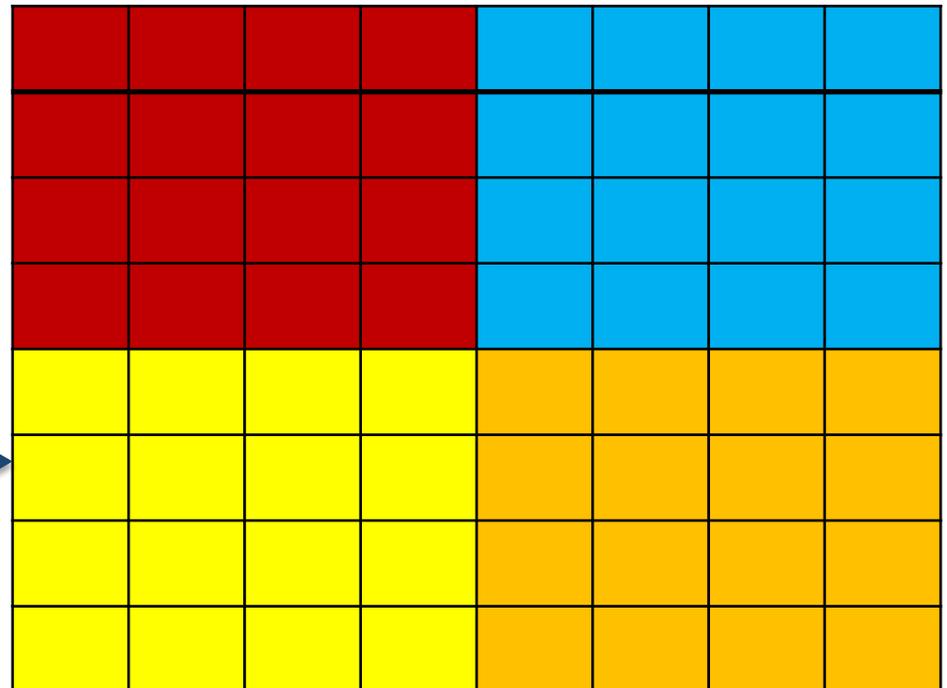
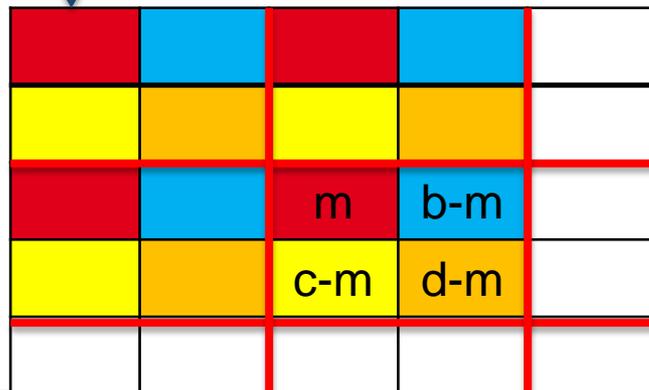
Exploitation de la dépendance des données

Cette transformation est un cas particulier :

- DCT transformée en cosinus discret (MPEG 1, 2, 4, futur)
- 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



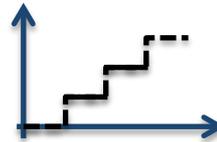
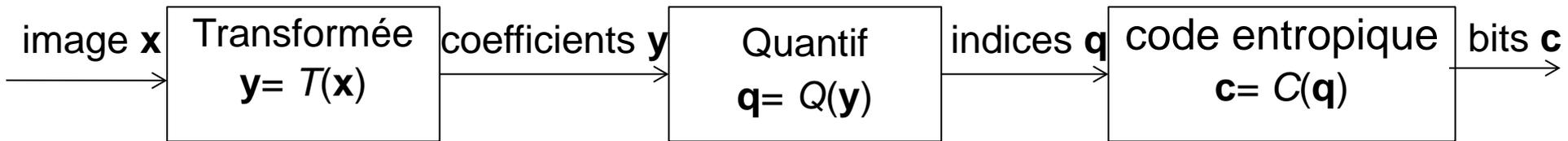
exemple d'ondelettes

Transformée utilisée dans JPEG 2000



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

JPEG2000



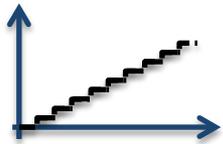
A	B	C	D
1/4	1/2	1/8	1/8
10	0	110	111

Résultats de compression JPEG2000



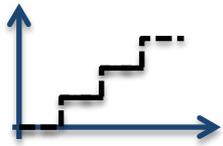
Original Carol (512 x 512 Pixels, 24-Bit RGB/pixel, Taille 786ko)

Résultats de compression JPEG2000



75:1, 10.6 ko

Résultats de compression JPEG2000



150:1, 5.3 ko

Résultats de compression JPEG2000



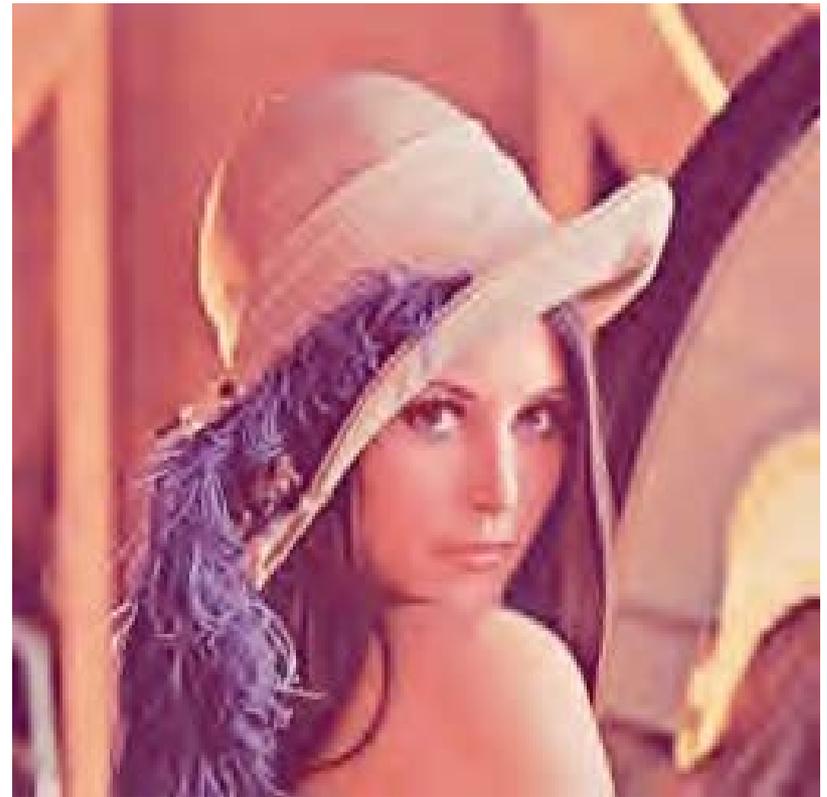
300:1, 2.6 ko



Comparaison JPEG (1992) vs. JPEG2000



Lena, 256x256 RGB
Baseline JPEG: 4572 octets



Lena, 256x256 RGB
JPEG-2000: 4572 octets

SOMMAIRE

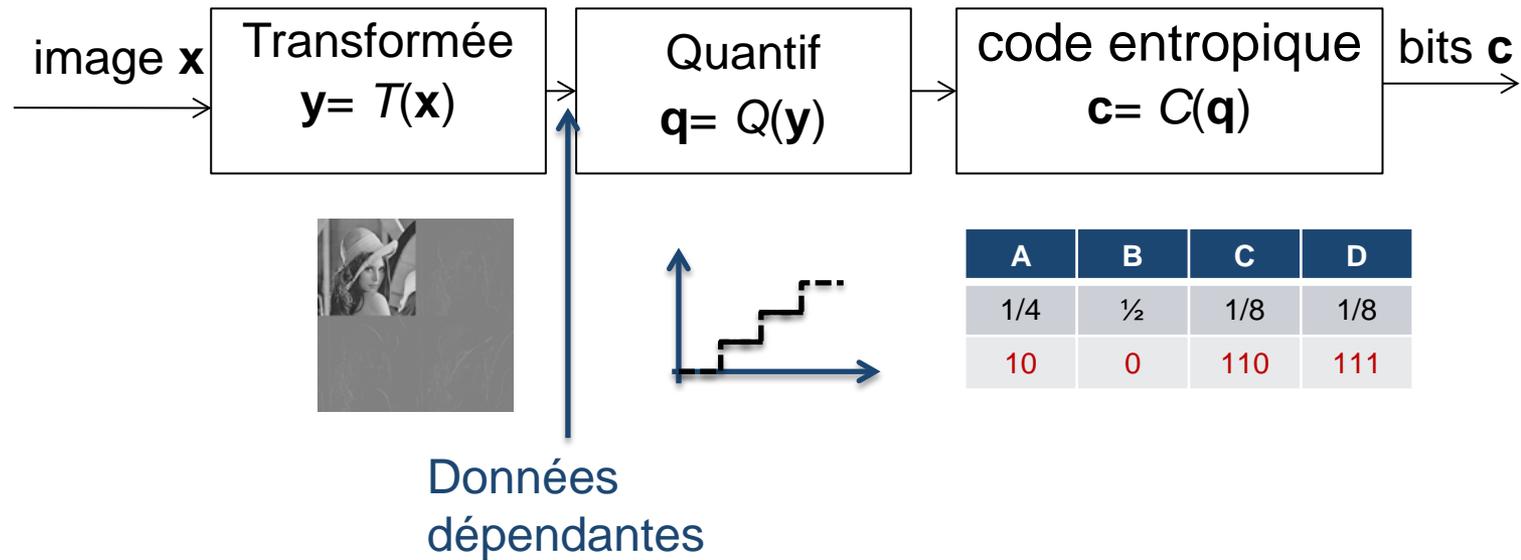
1. Pourquoi compresser une vidéo ?
2. Comment compresser ?
3. Compression avec pertes
4. Compression par transformation
5. **Compression par prédiction**

5

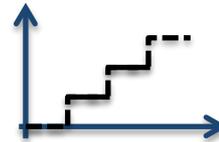
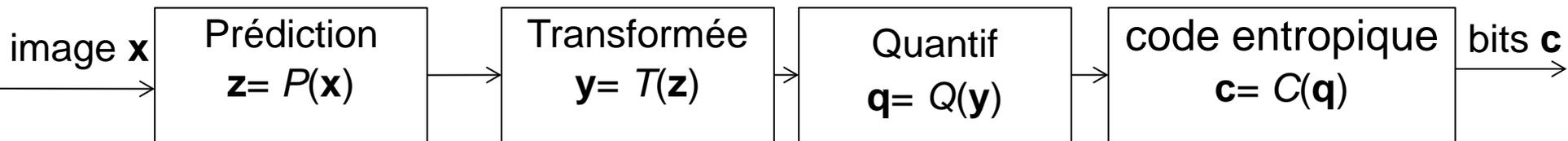
Données dépendantes

la prédiction

Il reste de la dépendance



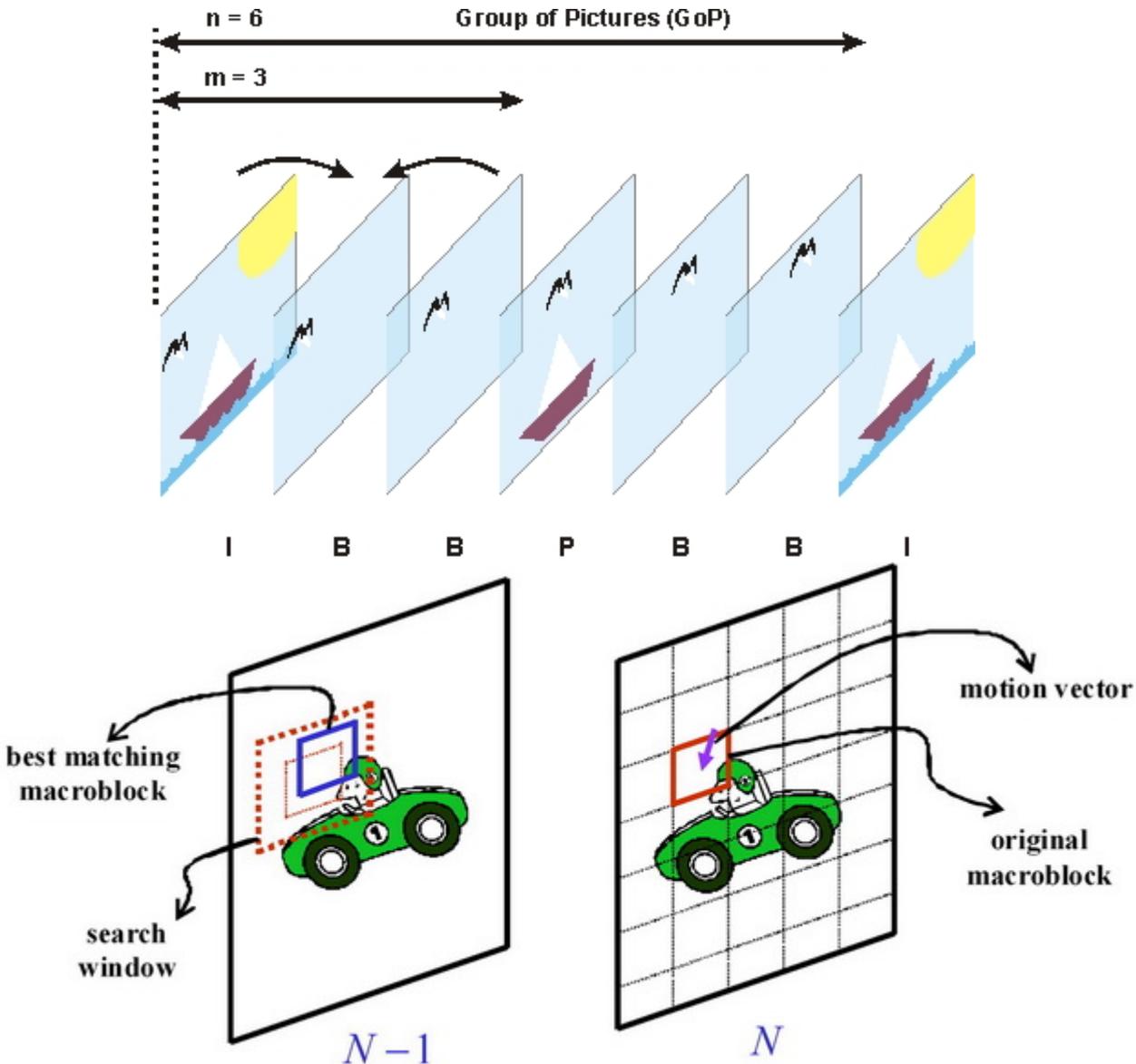
Il reste de la dépendance



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

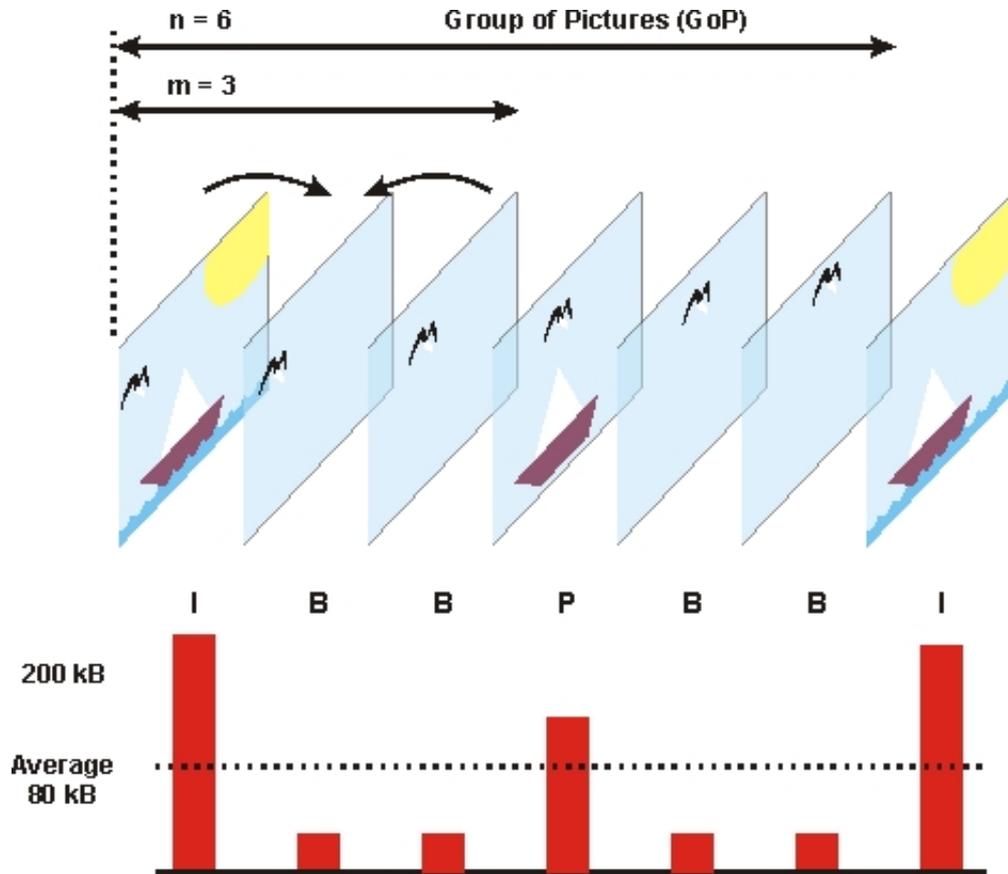


Intra: image codée indépendamment des autres (synchro)

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

Utilisé en MPEG-2, -4, HEVC

Prédiction... entre images



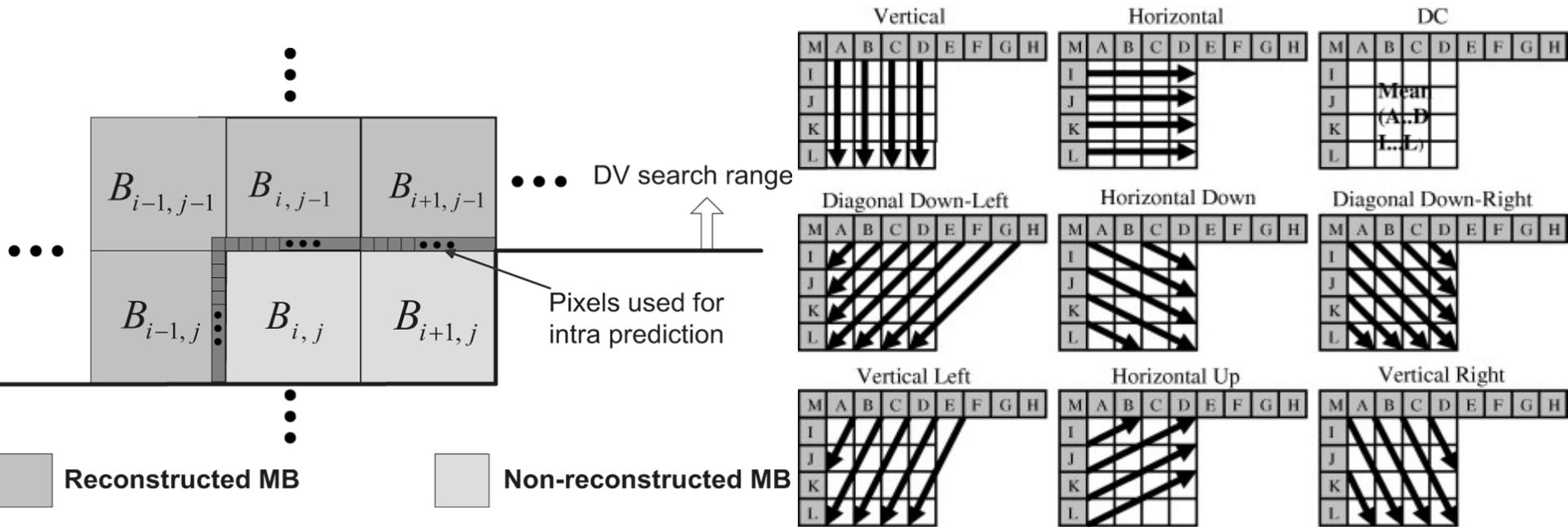
Intra: image codée indépendamment des autres (synchro)

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 (MPEG4, Hvc)



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

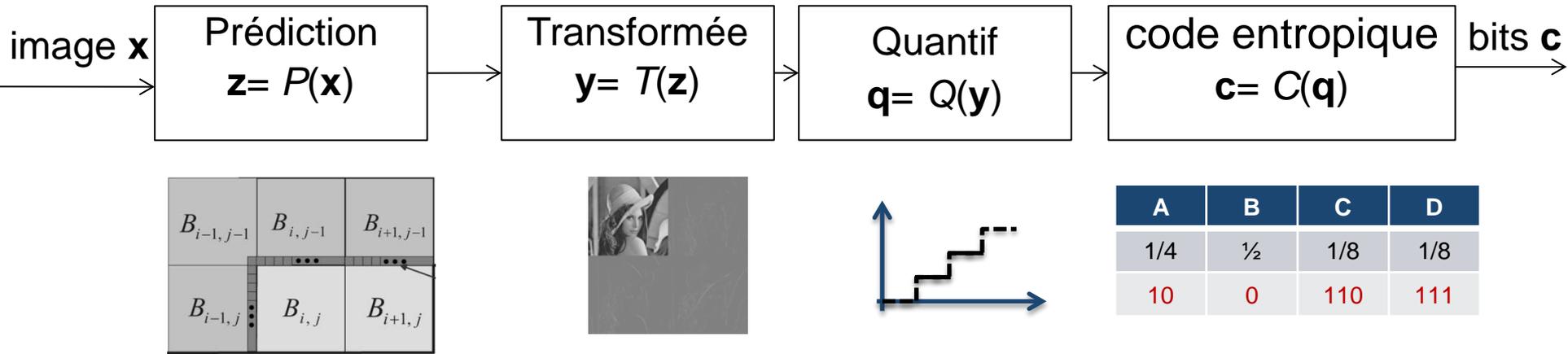
- échantillons indépendants [théorie de l'estimation]

6

Conclusion

les messages à ramener à la maison

Take-home-message



Transformation et prédiction : exploite dépendance des pixels

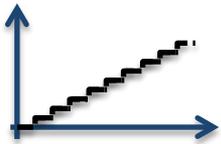
Quantification : introduit pertes

Codage entropique : exploite non égalité des probabilités des pixels

Quelques résultats en vidéo de MPEG-futur

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

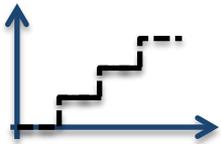


60:1

Quelques résultats en vidéo de MPEG-futur

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 MPEG-futur

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

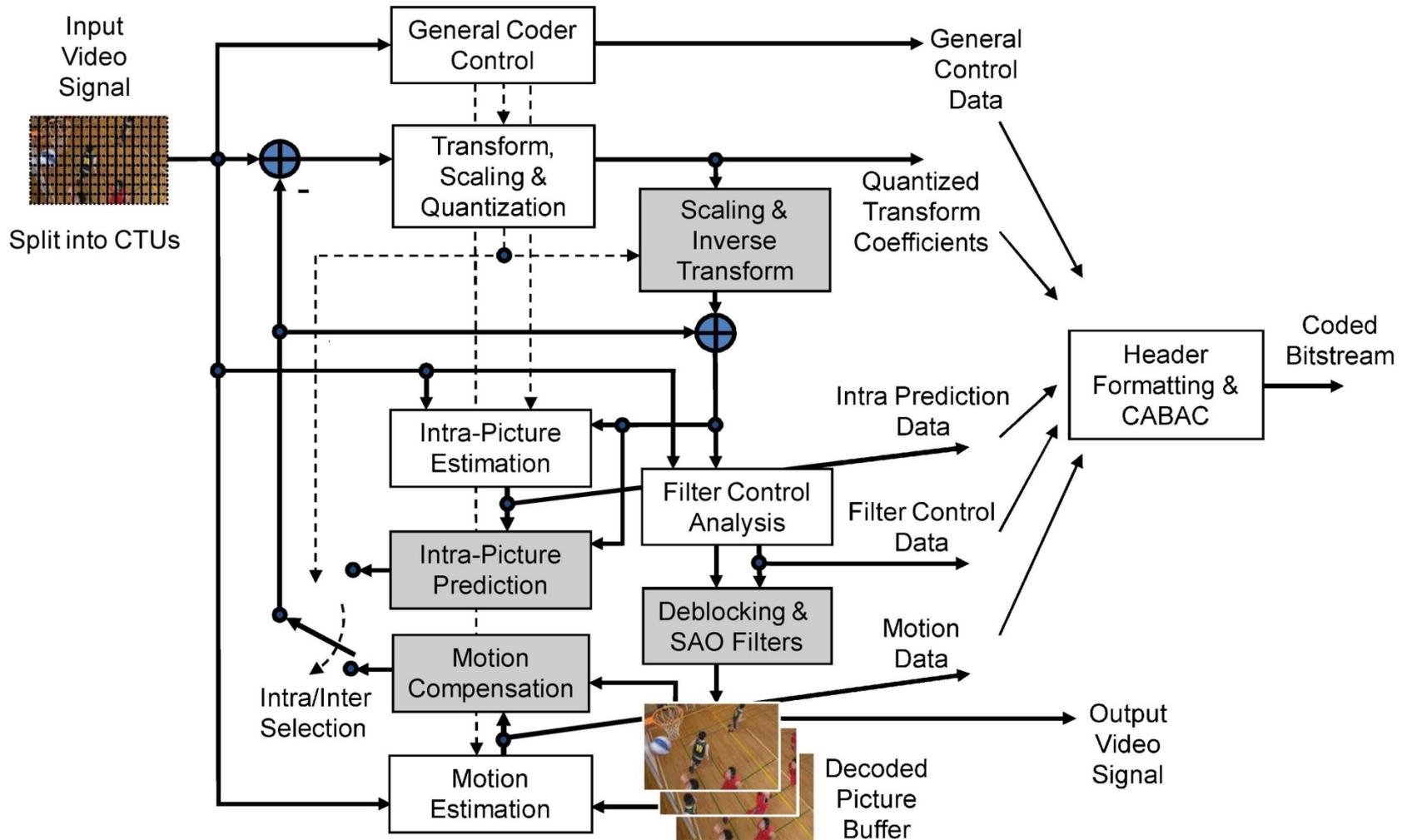


original



2000:1

Le schéma de codage HEVC



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

merci



www.inria.fr

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. Bernd Girod
Image and Video Compression EE398A, University of Stanford,
<http://www.stanford.edu/class/ee368/>

7

Conclusion

les mathématiques en compression

Soit $D \geq 0$. Soit $\sigma \geq 0$.

$$R(D) = \min_{f(x,y)} \int_{x,y} e^{-\frac{x^2}{2\sigma^2}} f(x,y) \log_2 \frac{f(x,y)}{e^{-\frac{x^2}{2\sigma^2}}} dx dy$$

avec contraintes

$$\forall x, y, f(x, y) \geq 0$$

$$\forall x, \int_y f(x, y) dy = 1$$

$$\int_{x,y} e^{-\frac{x^2}{2\sigma^2}} f(x, y) \|x - y\|^2 dx dy \leq \sigma \sqrt{2\pi} D$$

$$R(D) = \sigma^2 2^{-2R}$$

Soit $D \geq 0$. Soit p_1, \dots, p_N tels que $p_i \geq 0$ et $\sum_i p_i = 1$. Soit

$$f : \llbracket 1, N \rrbracket \rightarrow \mathbb{R}$$
$$i \mapsto f(i)$$

$$g : \llbracket 1, N \rrbracket \rightarrow \mathbb{R}$$
$$j \mapsto g(j)$$

$$R(D) = \min_{w_{i,j}} \sum_{i,j} w_{i,j} p_i \log_2 \frac{w_{i,j}}{p_i}$$

avec contraintes

$$w_{i,j} \geq 0$$

$$\forall i, \sum_j w_{i,j} = 1$$

$$\sum_{i,j} w_{i,j} p_i \|f(i) - g(j)\|^2 \leq D$$

merci



www.inria.fr