

POLYTECHNIQUE  
MONTRÉAL



LE GÉNIE  
EN PREMIÈRE CLASSE

NeuroPoly



# Compression et codage II

Eva Alonso Ortiz

ELE8812

30 mars 2023

# Plan

---

1. **Codage par transformée par blocs**
  - Démarche générale
  - Choix de la transformée
  - Sélection et quantification des coefficients
  - Redondance structurelle et codage des symboles
  - Exemple : codage JPEG
2. **Codage en ondelettes**
3. **Codage prédictif**
  - Codage prédictif sans et avec perte
  - Structure et spécification des prédicteurs
  - Quantification de l'erreur de prédiction
4. **Éléments de « tatouage numérique » (*watermarking*)**

# Plan

---

## 1. Codage par transformée par blocs

- Démarche générale
- Choix de la transformée
- Sélection et quantification des coefficients
- Redondance structurelle et codage des symboles
- Exemple : codage JPEG

## 2. Codage en ondelettes

## 3. Codage prédictif

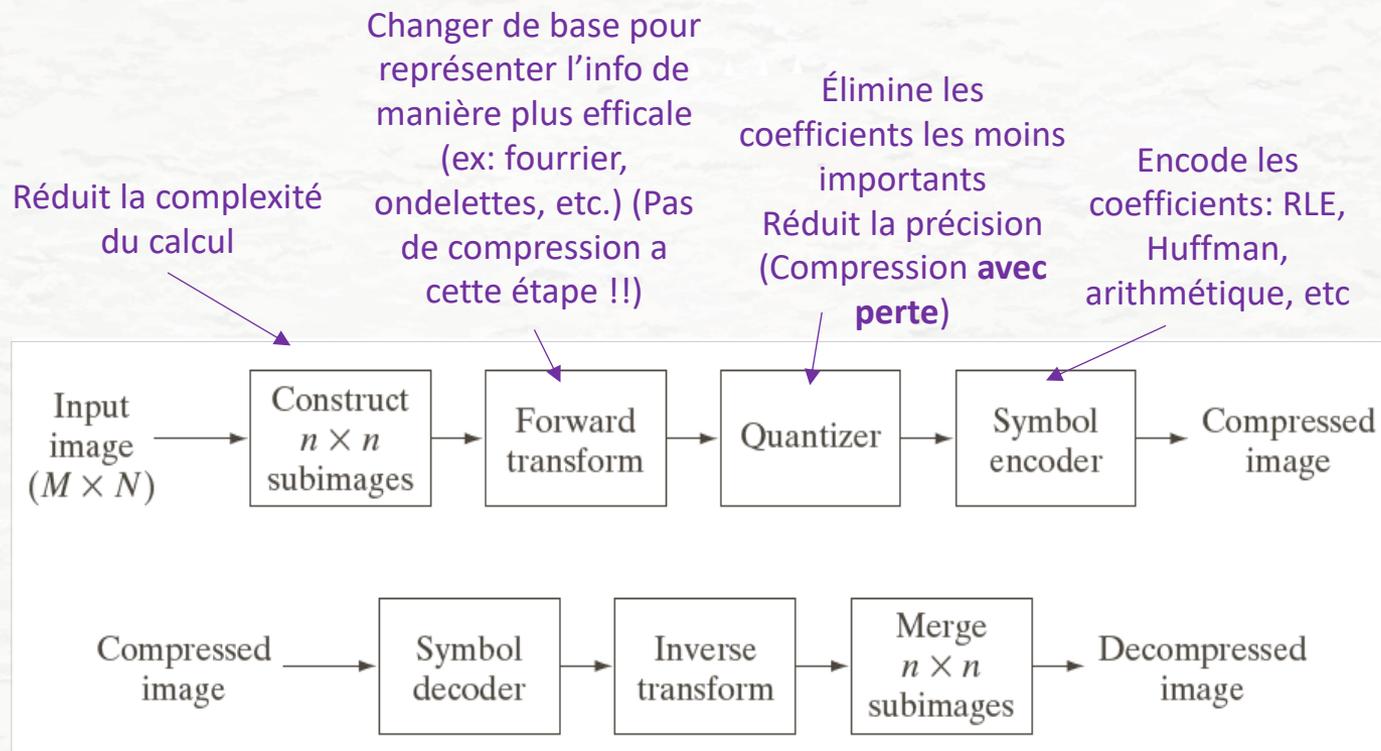
- Codage prédictif sans et avec perte
- Structure et spécification des prédicteurs
- Quantification de l'erreur de prédiction

## 4. Éléments de « tatouage numérique » (*watermarking*)

# Codage par transformée par blocs

## Démarche générale

### Étapes du codage et du décodage

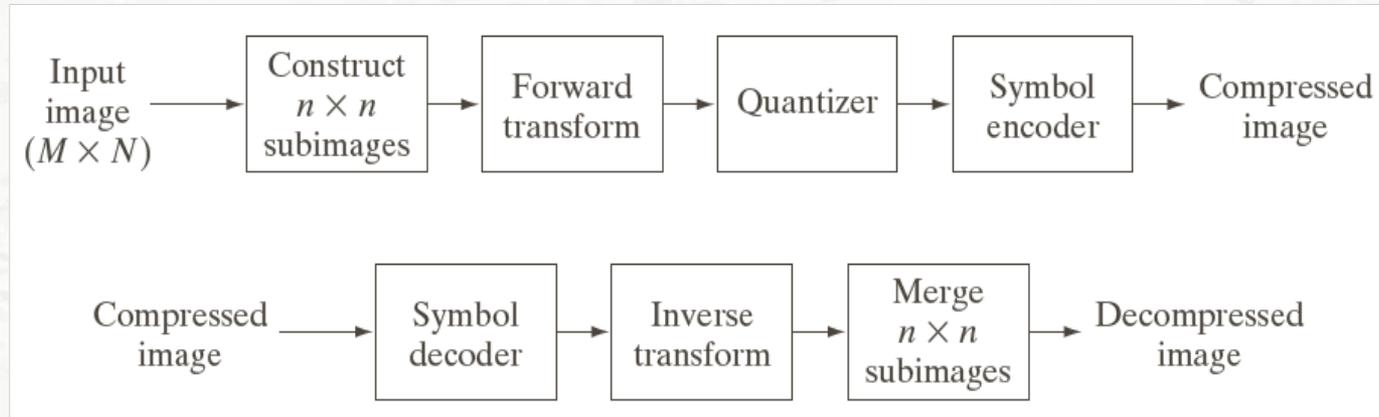


© 1992-2008 R. C. Gonzalez & R. E. Woods

# Codage par transformée par blocs

## Démarche générale

### Étapes du codage et du décodage



© 1992-2008 R. C. Gonzalez & R. E. Woods

### Points en suspens

- Choix de la taille des blocs
- Choix de la transformée
- Nature de la quantification
- Type de codage de symboles

# Transformés 2D et compression d'images

## Transformées discrètes usuelles

- En 2D: séparabilité
- Orthogonalité
- Signification physique (?)

Une transformée 2D est dite séparable si on peut la calculer ligne par ligne, puis colonne par colonne (réduit le cout de calcul) on évite les double intégrales/sommations.  
Ex: DFT, DCT, Transformée de Haar

Transformées de Fourier, de Walsh, en cosinus, Haar

## Fourier: Principales caractéristiques

- Calculs très rapides (algorithme FFT)
- Coefficients complexes
- Effets de bord significatifs dès qu'une troncature est effectuée
- Robustesse à la troncature et à la quantification (information packing ability) ?

# Transformés 2D et compression d'images

## Transformées discrètes usuelles

- En 2D: séparabilité
- Orthogonalité
- Signification physique (?)

Si les vecteurs de base de la transformée sont **orthogonaux** → chaque coefficient représente **une information indépendante**.  
→ **représentation compacte** → meilleure compression.

Transformées de Fourier, de Walsh, en cosinus, Haar

## Fourier: Principales caractéristiques

- Calculs très rapides (algorithme FFT)
- Coefficients complexes
- Effets de bord significatifs dès qu'une troncature est effectuée
- Robustesse à la troncature et à la quantification (information packing ability) ?

# Transformés 2D et compression d'images

## Transformées discrètes usuelles

- En 2D: séparabilité
- Orthogonalité
- Signification physique (?)

Espace vs fréquence ?  
Haar vs FFT ?

Transformées de Fourier, de Walsh, en cosinus, Haar

## Fourier: Principales caractéristiques

- Calculs très rapides (algorithme FFT) ✓
- Coefficients complexes, difficile à stocker ✗
- Effets de bord significatifs dès qu'une troncature est effectuée (artefacts visuels dans les régions contenant les bords/contours et haut contraste) ✗
- Robustesse à la troncature et à la quantification (information packing ability) ?

# Transformés 2D et compression d'images

## Définition

- Image  $f(x, y)$  ;  $0 \leq x \leq M - 1, 0 \leq y \leq N - 1$

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{2i\pi(ux/M + vy/N)}$$

- Pour chaque point  $f(x, y)$  il faut calculer une combinaison linéaire de fonctions 2D avec différentes fréquences évaluées au point  $(x, y)$

- Représenter  $f(x, y)$  comme matrice :

$$\mathbf{F} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u, v) \mathbf{S}_{u,v}$$

$\mathbf{F}$  : matrice  $N \times N$

$$\mathbf{S}_{u,v} = \begin{bmatrix} s(0,0,u,v) & s(0,1,u,v) & \dots & s(0,N-1,u,v) \\ s(1,0,u,v) & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ s(N-1,0,u,v) & s(N-1,1,u,v) & \dots & s(N-1,N-1,u,v) \end{bmatrix}$$

$T(u, v)$  = coefficient associé à la base  $\mathbf{S}_{u,v}$

$\mathbf{S}_{u,v}$  = fonction de base 2D associée à la fréquence  $u, v$

Pour  $u, v = 0, 1, \dots, N-1$

# Transformés 2D et compression d'images

Pour  $u, v = 0, 1, \dots, N-1$

$$\mathbf{F} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u, v) \mathbf{S}_{u, v}$$

$\mathbf{F}$  : matrice  $N \times N$

$$\mathbf{S}_{u, v} = \begin{bmatrix} s(0, 0, u, v) & s(0, 1, u, v) & \dots & s(0, N-1, u, v) \\ s(1, 0, u, v) & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ s(N-1, 0, u, v) & s(N-1, 1, u, v) & \dots & s(N-1, N-1, u, v) \end{bmatrix}$$

- $\mathbf{F}$  est une combinaison linéaire de  $N^2$  matrices de taille  $N \times N$
- $\mathbf{S}_{u, v}$  sont les *images de base*

$\mathbf{S}_{0,0}$	$\mathbf{S}_{0,1}$	...	...	$\mathbf{S}_{0,N-1}$
$\mathbf{S}_{1,0}$	$\ddots$			$\vdots$
$\vdots$				
			$\ddots$	
				$\vdots$
$\mathbf{S}_{N-1,0}$	...		...	$\mathbf{S}_{N-1,N-1}$

# Transformées 2D et compression d'images

$F \rightarrow 8 \times 8$

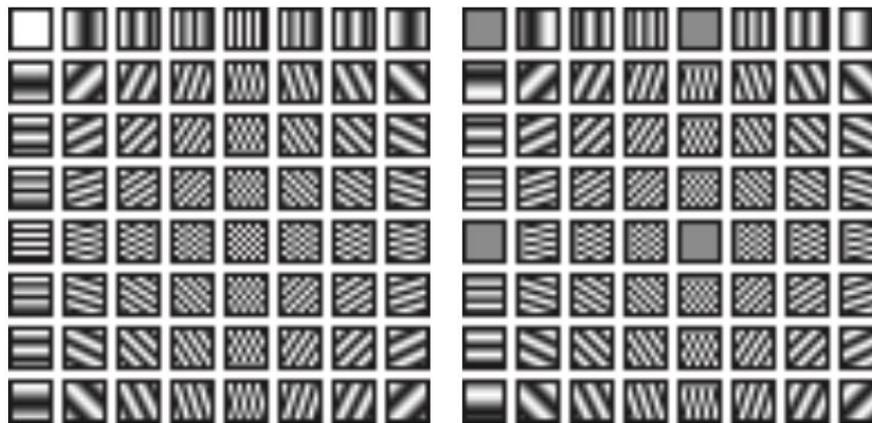
$$\mathbf{F} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u,v) \mathbf{S}_{u,v}$$

$F$  : matrice  $N \times N$

L'image du bloc  $F$  est une combinaison linéaire de 64 images de base, chacune ayant une structure fréquentielle différente

Images de base pour la TFD (partie réel/imaginaire)

Chaque  $S_{u,v}$  est une petite «brique 8 x 8» contenant un motif spécifique



Pourquoi DC en haut à gauche et non au milieu ?

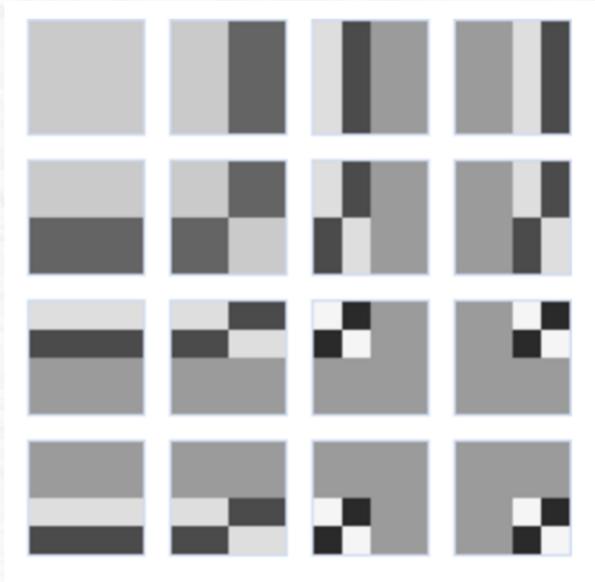
Constante (DC)

$\mathbf{s}_{0,0}$	$\mathbf{s}_{0,1}$	...	...	$\mathbf{s}_{0,N-1}$
$\mathbf{s}_{1,0}$	$\ddots$			$\vdots$
$\vdots$				
		$\ddots$		
$\mathbf{s}_{N-1,0}$	...		...	$\mathbf{s}_{N-1,N-1}$

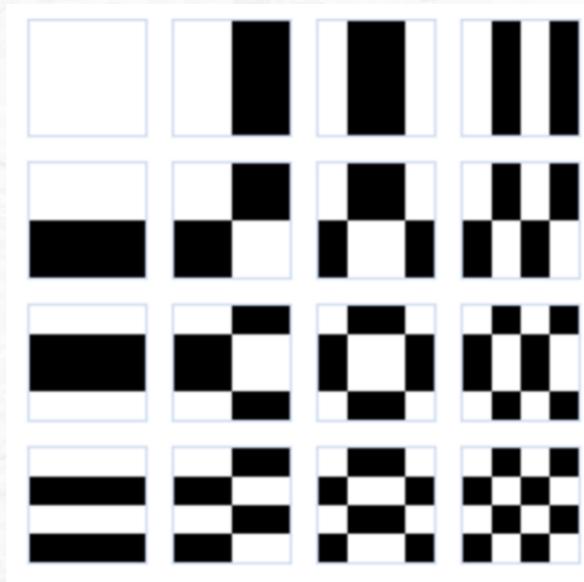
# Autres Transformées

 $F \rightarrow 4 \times 4$ 

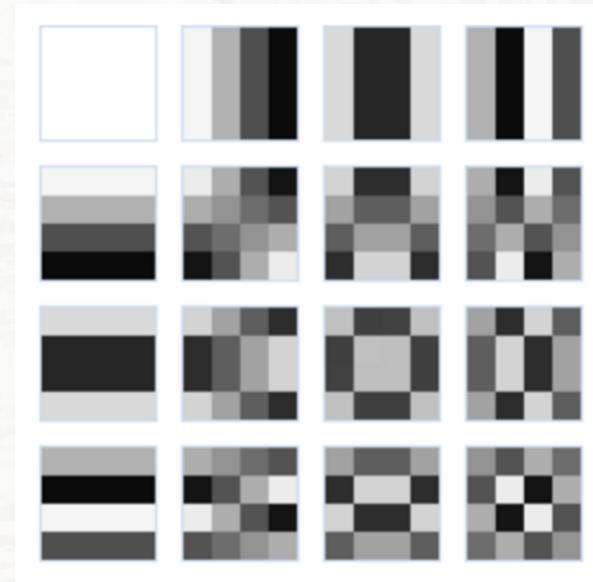
## Haar



## Walsh

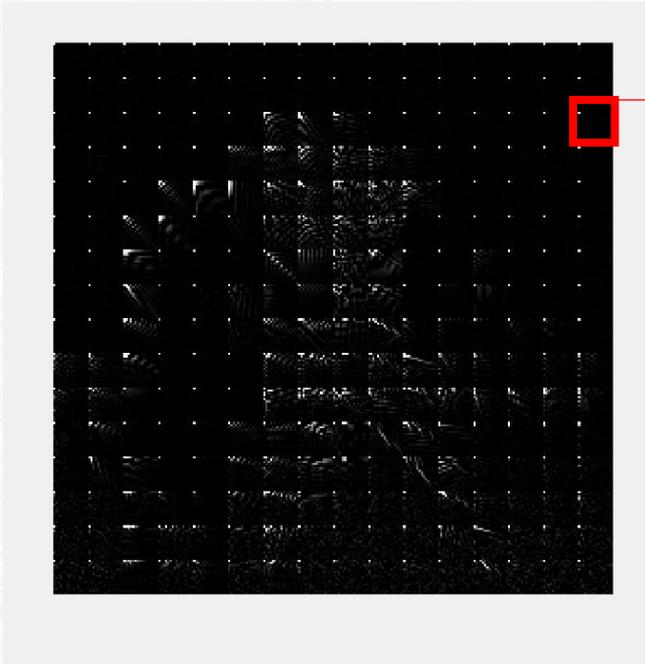


## Cosinus discrète



# TCD - Question

Image de taille 256 x 256, divisé en 256 blocs de taille 16x16



1 bloc de taille 16 x 16  
On calcule TCD pour  
chaque bloc  
Ce qui donne 16 x 16  
coefficients pour chaque  
bloc (256)

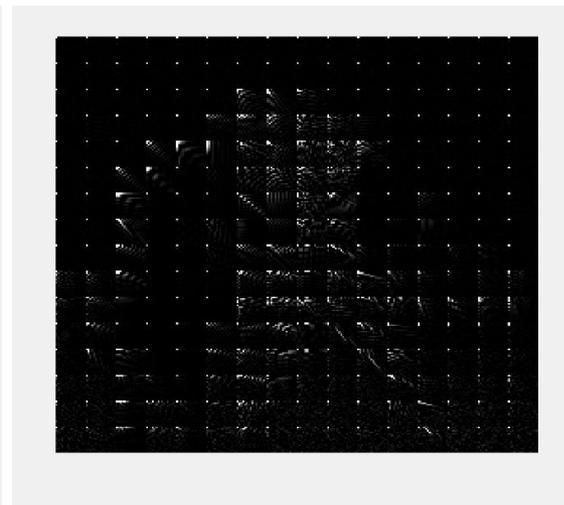
Pourquoi est-ce que on voit plus de structure/intensités dans les blocs centraux?

slido

Please download and install the Slido app on all computers you use



**Pourquoi est-ce que on voit plus de structure/intensités dans les blocs centraux?**



① Start presenting to display the poll results on this slide.

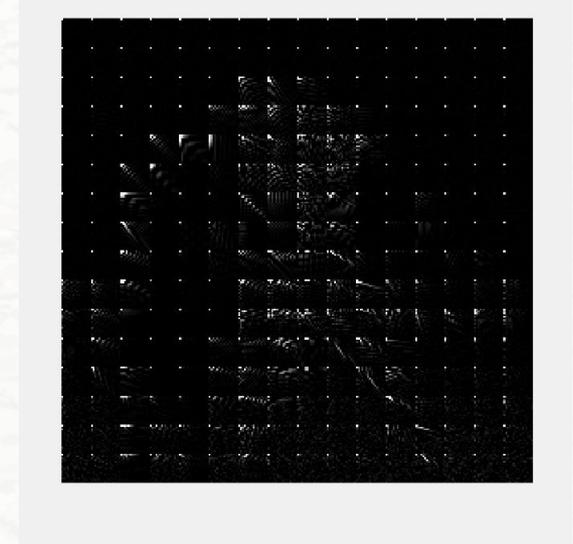
# TCD - Question



TCD sur l'image complète



TCD sur blocs de 16 x 16



Quel pourrait être un avantage de faire du codage des coefficients de la TCD sur blocs vs. codage des coefficients de la TCD sur l'image complète?

slido

Please download and install the Slido app on all computers you use



**Quel pourrait être un avantage de faire du codage des coefficients de la TCD sur bloques vs. codage des coefficients de la TCD sur l'image complète?**

① Start presenting to display the poll results on this slide.

# TCD - Question



Quel pourrait être un avantage de faire du codage des coefficients de la TCD sur blocs vs. codage des coefficients de la TCD sur l'image complète?

Même si la TCD globale donne plus de zéros, les coefficients restants sont peu redondants.

En TCD par blocs, les motifs de basse fréquence se répètent entre blocs → les coefficients sont redondants → meilleure compression avec un codage entropique.

# Transformées en cosinus discrète

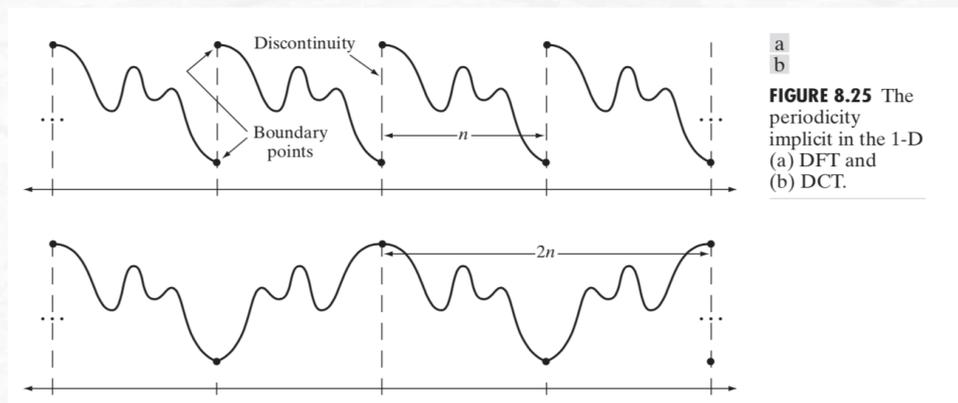
## Définition

- Cadre 1D : 
$$s_u(x) \propto \cos \left[ \left( x + \frac{1}{2} \right) \frac{u\pi}{n} \right]$$

- Cadre 2D 
$$s_{u,v}(x, y) = s_u(x)s_v(y)$$
 (séparabilité)

## Caractéristiques de la TCD

- Orthogonalité
- Simplicité des calculs  
(plus rapide que FFT)
- Coefficients reels ( stockage)
- Effets de bord réduits (elle suppose que le signal est périodique avec une symétrie paire autour des bords) moins de freq. haute inutiles → meilleure compression
- Amélioration de la robustesse à la troncature et à la quantification High "information packing ability"



**FIGURE 8.25** The periodicity implicit in the 1-D (a) DFT and (b) DCT.

# Comparaison des transformées

## Exemple : troncature de 50 %

Fourier



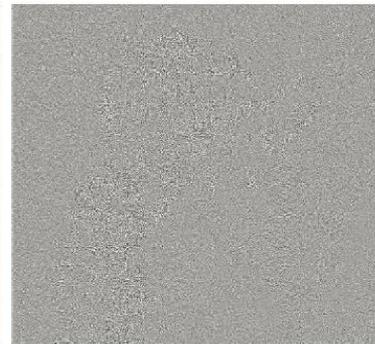
rms = 2.32

Walsh



rms = 1.78

Cosinus



rms = 1.13

# Sélection des coefficients

## Cadre général

- Décomposition : 
$$g(x, y) = \sum_{u,v=0}^{n-1} T(u, v) s_{u,v}(x, y)$$

- Décomposition tronquée : 
$$\hat{g}(x, y) = \sum_{u,v=0}^{n-1} T(u, v) \chi(u, v) s_{u,v}(x, y)$$

- Masque  $\chi(u, v)$  : compose de zéros (coefficients tronqués) et de uns

(coefficients sélectionnés) 
$$\chi(u, v) = \begin{cases} 1 & \rightarrow \text{on garde le coefficient} \\ 0 & \rightarrow \text{on le tronque (annule)} \end{cases}$$

Comment choisir  $\chi(u, v)$  ?

# Aspects pratiques

conservation des coefficients de plus forte amplitude (selection par seuil)

- On garde les coefficients  $T(u,v)$  les plus grands (en valeur absolue) peu importe leur position.
- Adaptatif (masque varie pour chaque sous-image)
- simple mais il faut stocker le masque pour chaque bloc

conservation des coefficients de plus forte variance (selection par zone)

- On garde les mêmes positions des coefficients  $T(u,v) \rightarrow$  estimer la variance Moyenne par position sur plusieurs imagelettes. Plus simple, moins d'overhead a stocker/transmettre

$X(7,12)$

1	1	0	1	<del>1</del>	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

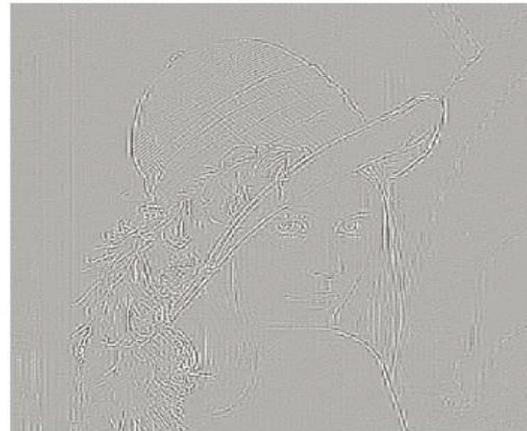
1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

# Aspects pratiques

## Exemple : seuil vs. zone



rms = 4.5



rms = 6.5

© 1992-2008 R. C. Gonzalez & R. E. Woods

# Sélection et quantification des symboles

## Approche mixte

- $\hat{T}(u, v) = \text{round} \left( \frac{T(u, v)}{Z(u, v)} \right)$
- *On divise chaque coefficient  $T(u, v)$  par un poids  $Z(u, v)$  puis on arrondi: perte contrôlée*
- $Z(u, v)$  déterminé à l'avance en fonction de l'impact visuel moyen des coefficients (petit pour les basses fréquences et grands pour les hautes)
- $Z(u, v)$  peut être multiplié par un coefficient de normalisation pour faire varier le taux de compression :  $Z'(u, v) = \text{coeff. de normalisation} \cdot Z(u, v)$

Plus le coeff. de normalisation est grand, plus on compresse, moins est la qualité

## Autres points

- Taille des blocs : 8 ou 16 ? Ex.: Le standard en JPEG est 8
- Ordonnancement 1D des coefficients
- Codage des symboles

# Exemple : codage JPEG (1)

---

## Principales caractéristiques (codage de base)

- 256 niveaux de gris
- Taille des blocs :  $n = 8$
- Utilisation de la TCD
- Quantification des coefficients avec une table de normalisation ( $Z(u,v)$  prédéfinie). 11 bits maximum
- Codage mixte des coefficients quantifiés (RLC (pour compacter les longues séries de zéros) + Huffman (pour compresser les symboles restants))
- Traitement particulier de  $\hat{T}(0,0)$  : DC: codé en différence avec le bloc précédent pour profiter de la redondance inter-bloc (car les valeurs moyennes changent lentement dans l'image)

## Exemple : codage JPEG (2)

Table de normalisation  $Z(u,v)$   
prédéterminée ! Tjrs la même!

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

© 1992-2008 R. C. Gonzalez & R. E. Woods

Ordonnancement des coefficients  
en zigzag

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

© 1992-2008 R. C. Gonzalez & R. E. Woods

# Étapes du codage

## 1. Image de départ [0, 255]

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	63	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

## 3. DCT appliquée sur un bloc normalisé

→ coefficients de fréquence

DC	-415	-29	-62	25	55	-20	-1	3
	7	-21	-62	9	11	-7	-6	6
	-46	8	77	-25	-30	10	7	-5
	-50	13	35	-15	-9	6	0	3
	11	-8	-13	-2	-1	1	-4	1
	-10	1	3	-3	-1	0	2	-1
	-4	-1	2	-1	2	-3	1	-2
	-1	-1	-1	-2	-1	-1	0	-1

## 2. Normalisation (« level shifting ») [-128, 127]

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

## 4. Quantification

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$\hat{T}(0,0) = \text{round} \left[ \frac{T(0,0)}{Z(0,0)} \right]$$

$$= \text{round} \left[ \frac{-415}{16} \right] = -26$$

Car la TCD est définie autour de zéro

# Étapes du codage

## Quantization

$$\hat{T}(0,0) = \text{round} \left[ \frac{T(0,0)}{Z(0,0)} \right]$$

$$= \text{round} \left[ \frac{-415}{16} \right] = -26$$

-26	-3	-6	2	2	0	0	0
1	2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

## Ordonnancement des coefficients

**[-26** -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

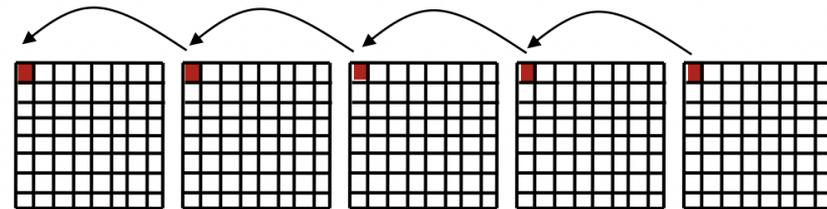
« End of Block »

## Construction du code JPEG (code DC)

- Différence entre le coefficient DC et celui du bloc précédent
- Ex: bloc précédent: -17

$$[-26 - (-17)] = -9$$

Pourquoi?



<https://web.stanford.edu/class/ee398a/handouts/lectures/08-JPEG.pdf>

# Étapes du codage

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

« End of Block »

## Construction du code JPEG (code DC)

- Correspondance entre différence et catégorie

Range	DC Difference Category
0	0
-1, 1	1
-3, -2, 2, 3	2
-7, ..., -4, 4, ..., 7	3
-15, ..., -8, 8, ..., 15	4
-31, ..., -16, 16, ..., 31	5
-63, ..., -32, 32, ..., 63	6
-127, ..., -64, 64, ..., 127	7
-255, ..., -128, 128, ..., 255	8
-511, ..., -256, 256, ..., 511	9
-1023, ..., -512, 512, ..., 1023	A
-2047, ..., -1024, 1024, ..., 2047	B
-4095, ..., -2048, 2048, ..., 4095	C
-8191, ..., -4096, 4096, ..., 8191	D
-16383, ..., -8192, 8192, ..., 16383	E
-32767, ..., -16384, 16384, ..., 32767	F

<https://web.stanford.edu/class/ee398a/handouts/lectures/08-JPEG.pdf>

# Étapes du codage

## Construction du code JPEG (code DC)

- Correspondance entre catégorie et mot de code

Category	Code length	Code word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

<https://web.stanford.edu/class/ee398a/handouts/lectures/08-JPEG.pdf>

En JPEG, les valeurs **négatives** sont représentées en inversant les bits du code binaire du positif

# Étapes du codage

---

## Construction du code JPEG (code DC)

- Pour une différence de +9 : 1001
- Pour une différence de -9 : 0110 (on inverse les bits)
- Catégorie 4 (slide précédente) → code word = 101
- Code complet : 1010110
- Mot de code complet : 7 bits

## Construction du code JPEG (code AC)

- Chaque code Huffman pour les coefficients AC dépend du
  - nombre de coefficients pas égales à zéro qui précède le prochain coefficient non nul
  - La catégorie de magnitude

# Étapes du codage

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

## Construction du code JPEG (code AC)

- Ex: -3 sans coefficients précédents. (Runlength, size) = (0,2)
- Catégorie 2

Range	DC Difference Category	AC Category
0	0	N/A
-1, 1	1	1
-3, -2, 2, 3	2	2
-7, ..., -4, 4, ..., 7	3	3
-15, ..., -8, 8, ..., 15	4	4
-31, ..., -16, 16, ..., 31	5	5
-63, ..., -32, 32, ..., 63	6	6
-127, ..., -64, 64, ..., 127	7	7
-255, ..., -128, 128, ..., 255	8	8
-511, ..., -256, 256, ..., 511	9	9
-1023, ..., -512, 512, ..., 1023	A	A
-2047, ..., -1024, 1024, ..., 2047	B	B
-4095, ..., -2048, 2048, ..., 4095	C	C
-8191, ..., -4096, 4096, ..., 8191	D	D
-16383, ..., -8192, 8192, ..., 16383	E	E
-32767, ..., -16384, 16384, ..., 32767	F	N/A

<https://web.stanford.edu/class/ee398a/handouts/lectures/08-JPEG.pdf>

# Étapes du codage

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

## Construction du code JPEG (code AC)

- Ex: -3 sans coefficients précédents (0,2)
  - Catégorie 2
  - 01 (sans bit précédents)
  - Length = 4 donc 2 bits manquants
  - 3 en binaire = 11
  - -3 on inverse: 00
  - 0100

Run/Category	Base Code	Length	Run/Category	Base Code	Length
0/0	1010 (= EOB)	4			
0/1	00	3	8/1	11111010	9
0/2	01	4	8/2	11111111000000	17
0/3	100	6	8/3	111111110110111	19
0/4	1011	8	8/4	111111110111000	20
0/5	11010	10	8/5	111111110111001	21
0/6	111000	12	8/6	111111110111010	22
0/7	1111000	14	8/7	111111110111011	23
0/8	111110110	18	8/8	111111110111100	24
0/9	111111110000010	25	8/9	111111110111101	25
0/A	1111111110000011	26	8/A	111111110111110	26
1/1	1100	5	9/1	11111000	10
1/2	111001	8	9/2	111111110111111	18
1/3	1111001	10	9/3	111111111000000	19
1/4	111110110	13	9/4	111111111000001	20
1/5	11111110110	16	9/5	111111111000010	21
1/6	111111110000100	22	9/6	111111111000011	22
1/7	1111111110000101	23	9/7	1111111110000100	23
1/8	1111111110000110	24	9/8	1111111110000101	24
1/9	1111111110000111	25	9/9	1111111110000110	25
1/A	11111111100001000	26	9/A	1111111110000111	26
2/1	11011	6	A/1	111111001	10
2/2	11111000	10	A/2	1111111111001000	18
2/3	1111110111	13	A/3	1111111111001001	19
2/4	111111110001001	20	A/4	1111111111001010	20
2/5	1111111110001010	21	A/5	1111111111001011	21
2/6	1111111110001011	22	A/6	1111111111001100	22
2/7	1111111110001100	23	A/7	1111111111001101	23

<https://web.stanford.edu/class/ee398a/handouts/lectures/08-JPEG.pdf>

# Étapes du codage

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]



1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001  
001 100101 11100110 110110 0110 11110100 000 1010

Compression:  $\frac{8 \times 8 \times 8}{92} \approx 5.6$

# Étapes du décodage

---

```
1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001  
001 100101 11100110 110110 0110 11110100 000 1010
```



```
[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]
```

# Étapes du décodage

## 1. Coefficients Quantifiés

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

## 3. Inverse DCT

-70	-64	-61	-64	-69	-66	-58	-50
-72	-73	-61	-39	-30	-40	-54	-59
-68	-78	-58	-9	13	-12	-48	-64
-59	-77	-57	0	22	-13	-51	-60
-54	-75	-64	-23	-13	-44	-63	-56
-52	-71	-72	-54	-54	-71	-71	-54
-45	-59	-70	-68	-67	-67	-61	-50
-35	-47	-61	-66	-60	-48	-44	-44

*-128 → 127*

## 2. De-quantification $\hat{T}(u, v) = \hat{T}(u, v)Z(u, v)$

-416	-33	-60	32	48	0	0	0
12	-24	-56	0	0	0	0	0
-42	13	80	-24	-40	0	0	0
-56	17	44	-29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

## 4. De-normalisation (« level shifting »)

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

# Étapes du décodage

## Image de départ

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	63	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

## Réconstruction

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

## Difference

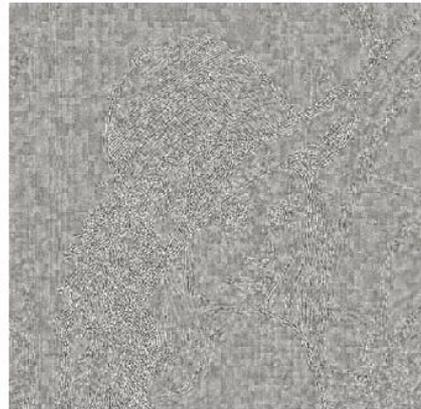
-6	-9	-6	2	11	-1	-6	-5
7	4	-1	1	11	-3	-5	3
2	9	-2	-6	-3	-12	-14	9
-6	7	0	-4	-5	-9	-7	1
-7	8	4	-1	6	4	3	-2
3	8	4	-4	2	6	1	1
2	2	5	-1	-6	0	-2	5
-6	-2	2	6	-4	-4	-6	10

RMSE = 5.8

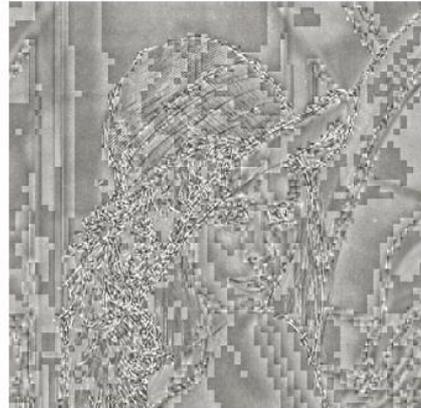
# Codage JPEG

Exemple de résultat

## Taux de compression de 25 et de 52



RMS = 5.4



RMS = 10.7

© 1992-2008 R. C. Gonzalez & R. E. Woods

# Plan

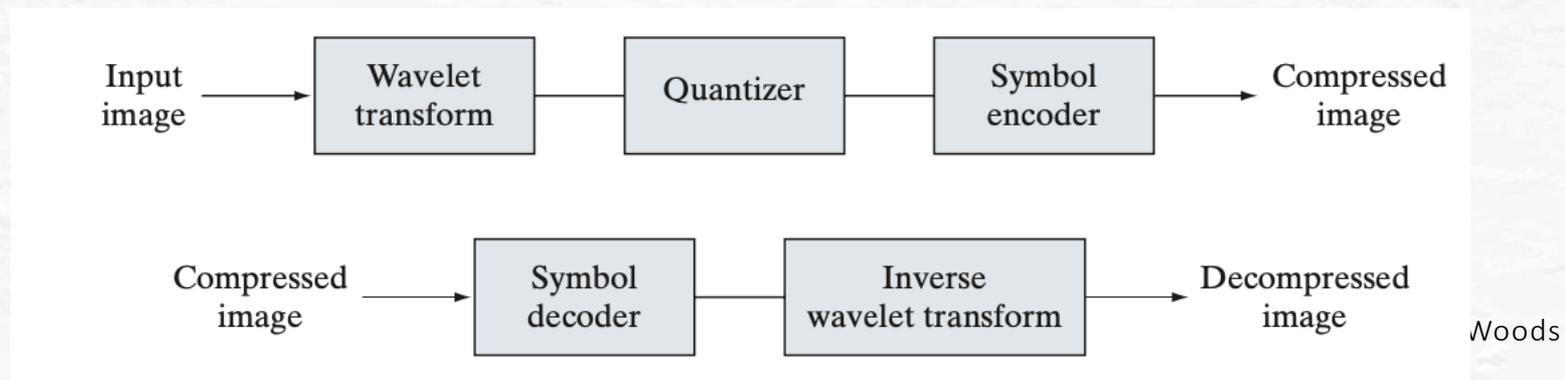
---

1. **Codage par transformée par blocs**
  - Démarche générale
  - Choix de la transformée
  - Sélection et quantification des coefficients
  - Redondance structurelle et codage des symboles
  - Exemple : codage JPEG
2. **Codage en ondelettes**
3. **Codage prédictif**
  - Codage prédictif sans et avec perte
  - Structure et spécification des prédicteurs
  - Quantification de l'erreur de prédiction
4. **Éléments de « tatouage numérique » (*watermarking*)**

# Codage en ondelettes (1)

## Démarche générale

- Transformation : coefficients horizontaux, verticaux, diagonaux
  - Transformée en ondelettes discrète ou transformée en ondelette rapide
- Quantification et codage des coefficients



## Caractéristiques

- Pas de sous-images : pas de « blocking artifact » caractéristique du DCT

# Codage en ondelettes (2)

---

## Points en suspens

- Choix d'ondelette (Haar, Daubechies, Symlet, Biorthogonal)
  - Complexité des calculs, « information packing ability »
- Niveaux de décomposition (transformée en ondelettes rapide)
  - Nombre d'opérations
- Choix d'approche pour la quantification

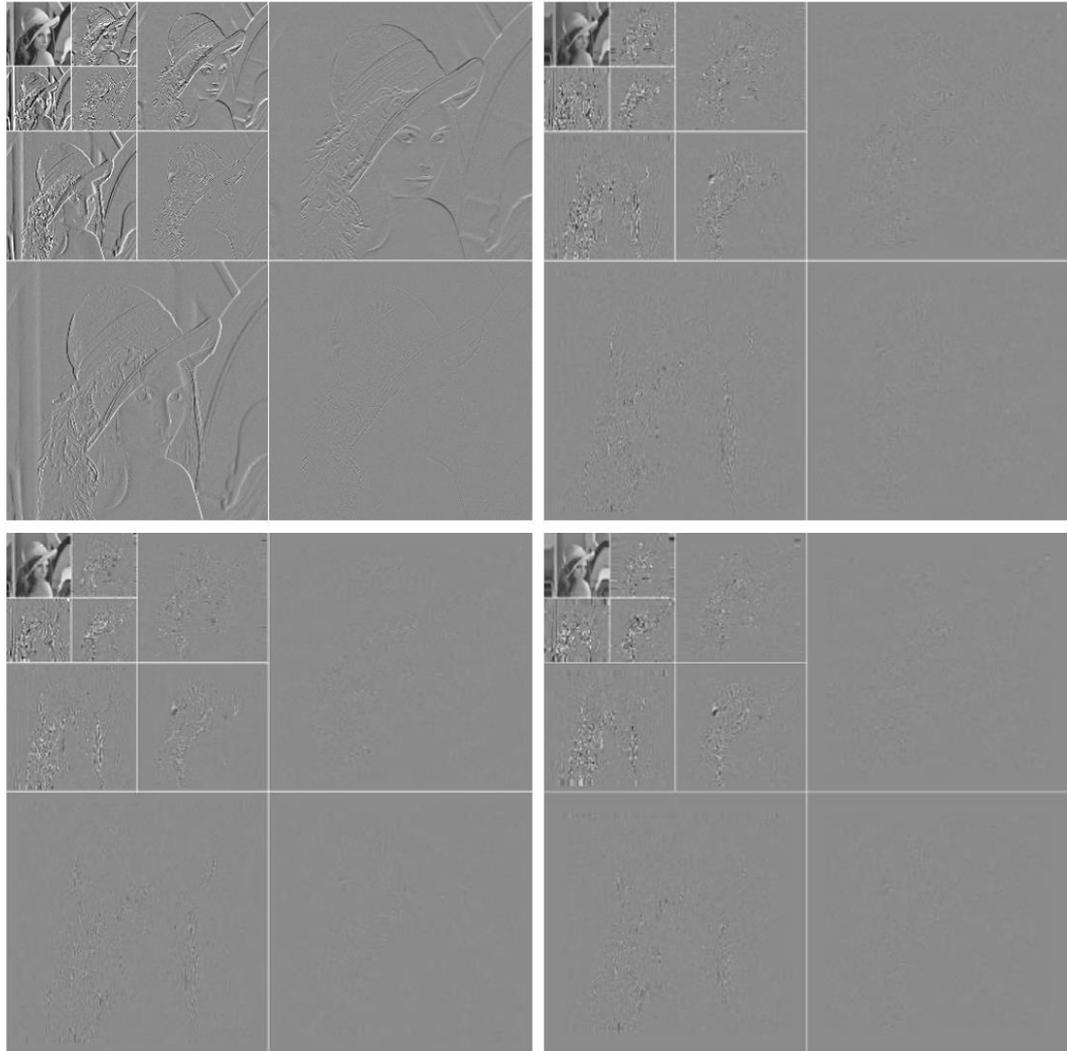
Demo: [1DSignalDecomposition.ipynb](#)

# Codage en ondelettes (3)

a b  
c d

**FIGURE 8.43**

Three-scale wavelet transforms of Fig. 8.9(a) with respect to (a) Haar wavelets, (b) Daubechies wavelets, (c) symlets, and (d) Cohen-Daubechies-Feauveau biorthogonal wavelets.

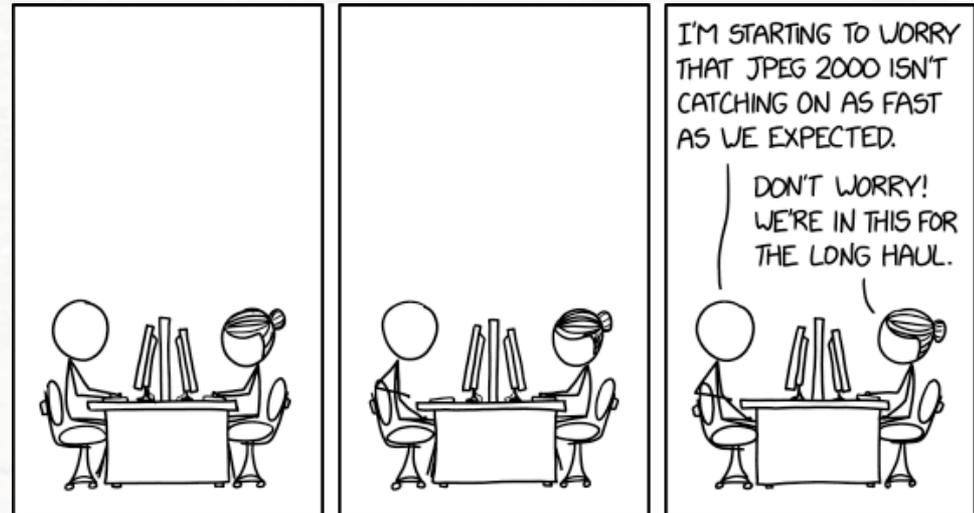


© 1992-2008 R. C. Gonzalez & R. E. Woods

# Codage en ondelettes (4)

## Application : JPEG-2000

- Développé vers la fin du 20<sup>ième</sup> siècle
- Version optimisée du JPEG
  - Versatile : compression avec et sans perte d'information
  - Meilleure qualité d'image
  - Plus haut taux de compression
- JPEG demeure le standard ....



# Codage en ondelettes (5)

---

## Application : JPEG-2000

- LizardTech : Brevet sur tout le codage a base d'ondelettes
  - Brevet invalidé
- Microsoft : pas de support (propre standard JPEG XR)
- Pas de support des navigateurs et du matériel informatique
- Performance : lent
- Améliorations récentes
- Usage pour les images médicales

# Plan

---

1. **Codage par transformée par blocs**
  - Démarche générale
  - Choix de la transformée
  - Sélection et quantification des coefficients
  - Redondance structurelle et codage des symboles
  - Exemple : codage JPEG
2. **Codage en ondelettes**
3. **Codage prédictif**
  - Codage prédictif sans et avec perte
  - Structure et spécification des prédicteurs
  - Quantification de l'erreur de prédiction
4. **Éléments de « tatouage numérique » (*watermarking*)**

# Codage prédictif (1)

---

## Principe

- Réduire des redondances spatiales/temporelles de pixels proches
- Coder des changements
- Image composée d'une partie certaine (modèle déterministe) et d'une partie incertaine (déviation par rapport au modèle)

$$f(n) = e(n) + \hat{f}(n)$$

$\hat{f}(n)$ : valeur prédite du pixel  $n$

$e(n)$ : l'erreur de prédiction

- Stockage
  - des coefficients du modèle
  - de la déviation par rapport au modèle  $e(n)$
  - Si la différence est faible, on peut coder avec peu de bits

# Codage prédictif (2)

---

## Codage prédictif

- Pixel courant : combinaison linéaire des  $m$  pixels *passés*

$$\hat{f}(n) = \text{round} \left[ \sum_{i=1}^m \alpha_i f(n - i) \right]$$

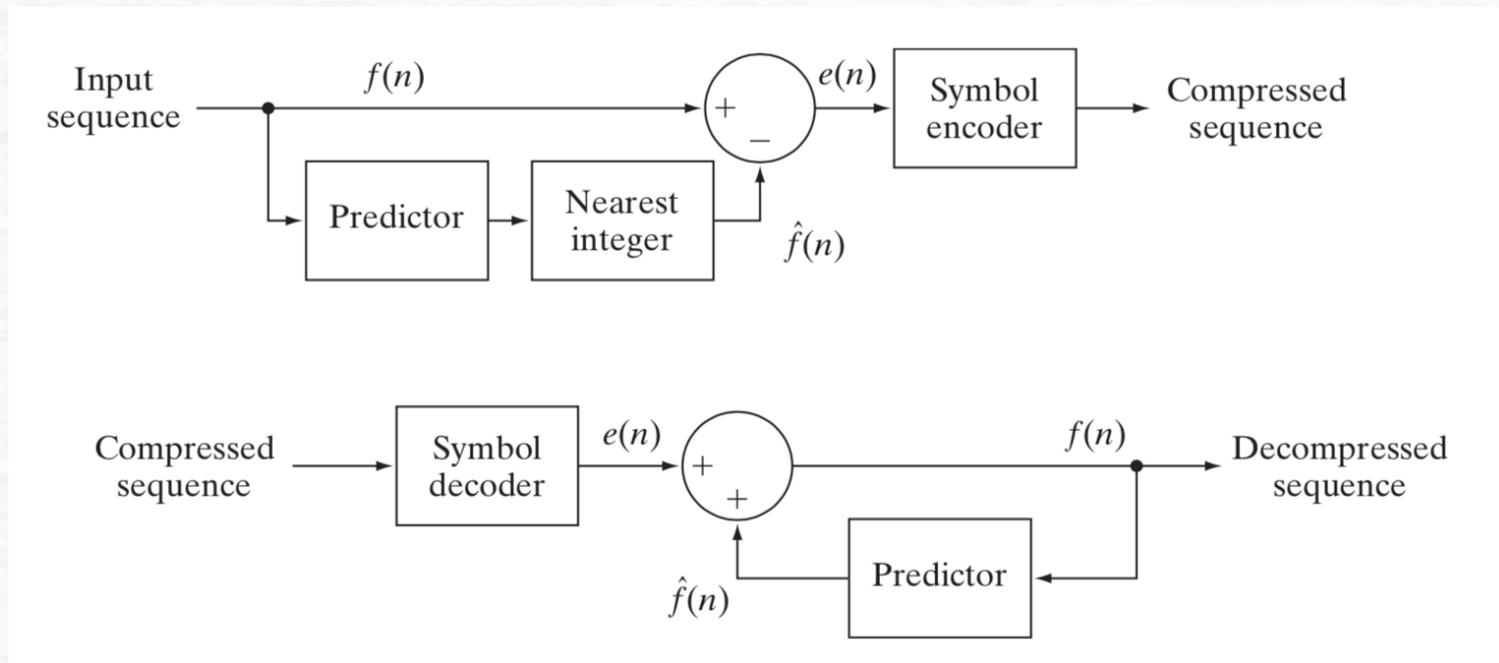
$$e(n) = f(n) - \hat{f}(n)$$

## Codage sans ou avec pertes

- Sans perte : déviation stockée exactement
- Avec pertes : déviation quantifiée et stockée avec approximation

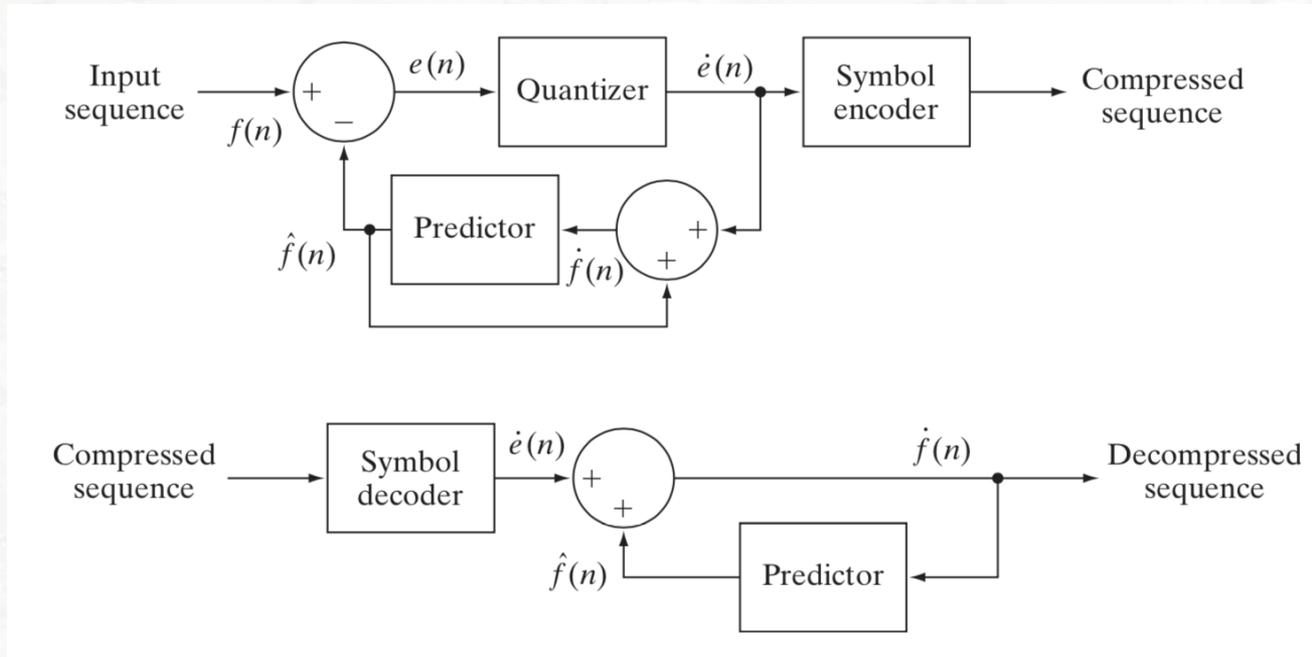
# Codage prédictif (3)

## Structure du codeur et du décodeur (sans perte)



# Codage prédictif (4)

## Structure du codeur et du décodeur (avec perte)



### Différence concrète entre avec/sans perte ?

→ la **quantification de l'erreur** : avec perte, on simplifie les valeurs d'erreur pour mieux les compresser.

© 1992-2008 R. C. Gonzalez & R. E. Woods

# Structure des prédicteurs (1)

---

## Images fixes

- Prédiction du pixel courant à partir des **pixels précédents** les plus proches (1, 2, 3 ou mixte)
  - ex. 1D:  $\hat{f}(x, y) = \text{round}[\alpha f(x, y - 1)]$
  - ex 2D:  $\hat{f}(x, y) = \alpha_1 f(x, y - 1) + \alpha_2 f(x - 1, y) + \alpha_3 f(x - 1, y - 1)$
- Transmission des  $\{\alpha_j\}$  et de l'erreur de prédiction (quantifiée ou non)

## Séquence d'images (Vidéo)

- Prédiction du pixel courant à partir du (des) pixel(s) correspondant(s) de **l'image précédente**
- Estimation locale des mouvements ("Motion Compensation")

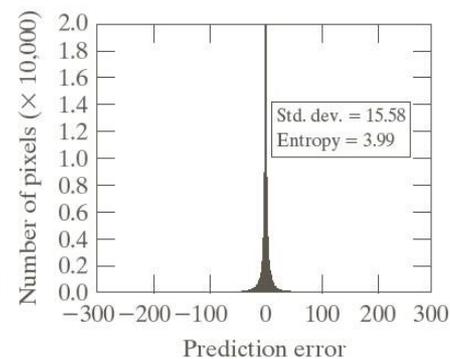
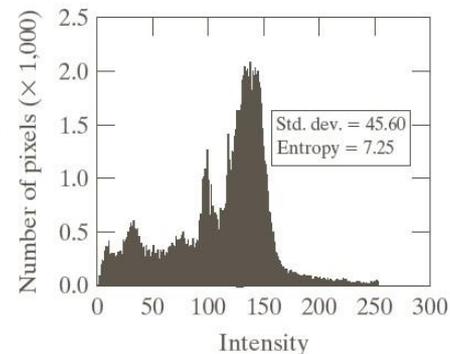
# Exemple

## Image vs. déviation par rapport au modèle

- Codage prédictif 1D : « previous pixel coding »

$$\hat{f}(x, y) = \text{round}[\alpha f(x, y - 1)]$$

$$\alpha = 1$$



a b  
c d

**FIGURE 8.34**  
(a) A view of the Earth from an orbiting space shuttle. (b) The intensity histogram of (a). (c) The prediction error image resulting from Eq. (8.2-34). (d) A histogram of the prediction error. (Original image courtesy of NASA.)

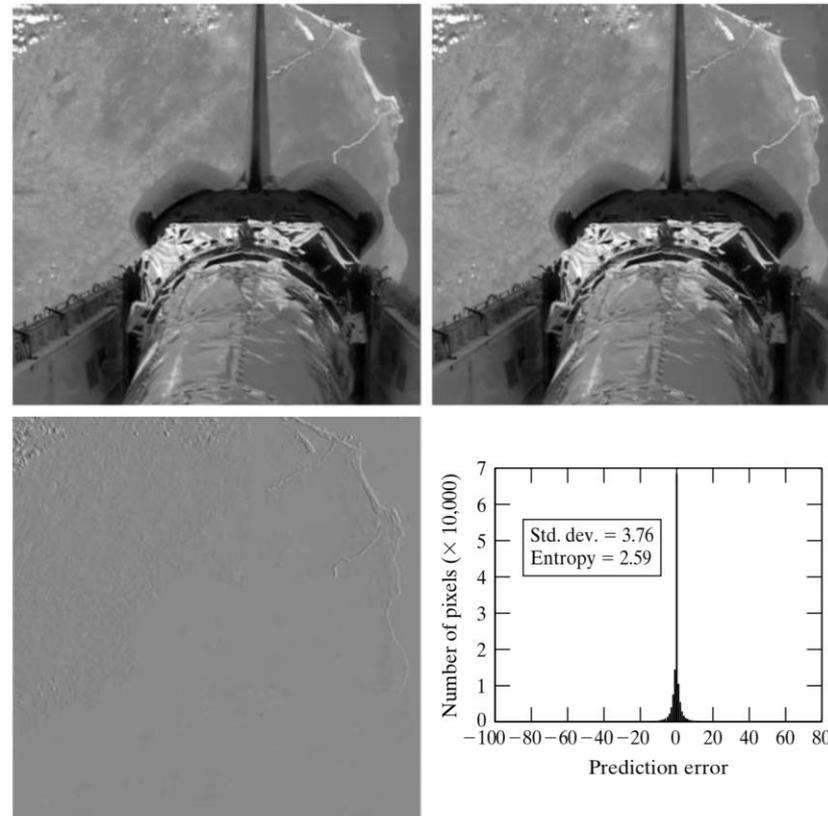
# Exemple

## Image vs. déviation par rapport au modèle

- Codage prédictif 1D : « previous pixel coding »

$$\hat{f}(x, y, t) = \text{round}[\alpha f(x, y, t - 1)]$$

$$\alpha = 1$$



a b  
c d

**FIGURE 8.35**  
(a) and (b) Two views of Earth from an orbiting space shuttle video. (c) The prediction error image resulting from Eq. (8.2-36). (d) A histogram of the prediction error. (Original images courtesy of NASA.)

# Exemple

---

$$\hat{f}(x, y) = 0.97f(x, y - 1) \quad (8.2-53)$$

$$\hat{f}(x, y) = 0.5f(x, y - 1) + 0.5f(x - 1, y) \quad (8.2-54)$$

$$\hat{f}(x, y) = 0.75f(x, y - 1) + 0.75f(x - 1, y) - 0.5f(x - 1, y - 1) \quad (8.2-55)$$



# Spécification des coefficients de prédiction

## Spécification *a priori*

- Spécification empirique des paramètres
- Choix Classique :  $f(x, y) = f(x, y - 1) + \epsilon(x, y)$

## Estimation

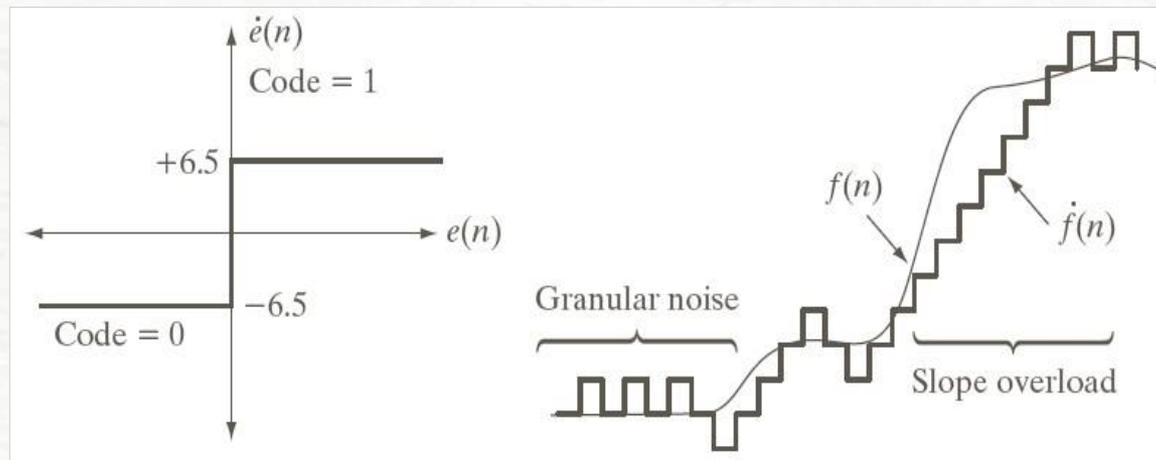
- Prédiction linéaire  $f = F \alpha + \epsilon$
- $F$  : matrice construite à partir des pixels de l'image
- $\alpha$  : vecteur des coefficients à estimer
- $\epsilon$  : erreur de prédiction
- Estimateurs classiques avec forme explicite (par ex., moindres carrés)
  - Ex: moindres carrés:  $\hat{\alpha} = (F^T F)^{-1} F^T f$

# Quantification de l'erreur de prédiction

## Un choix simple : la modulation delta (avec perte)

- $\epsilon(x, y) \geq 0 \Rightarrow \dot{\epsilon}(x, y) = \zeta$
- $\epsilon(x, y) < 0 \Rightarrow \dot{\epsilon}(x, y) = -\zeta$
- Quantification de l'erreur de prédiction avec 1 bit !

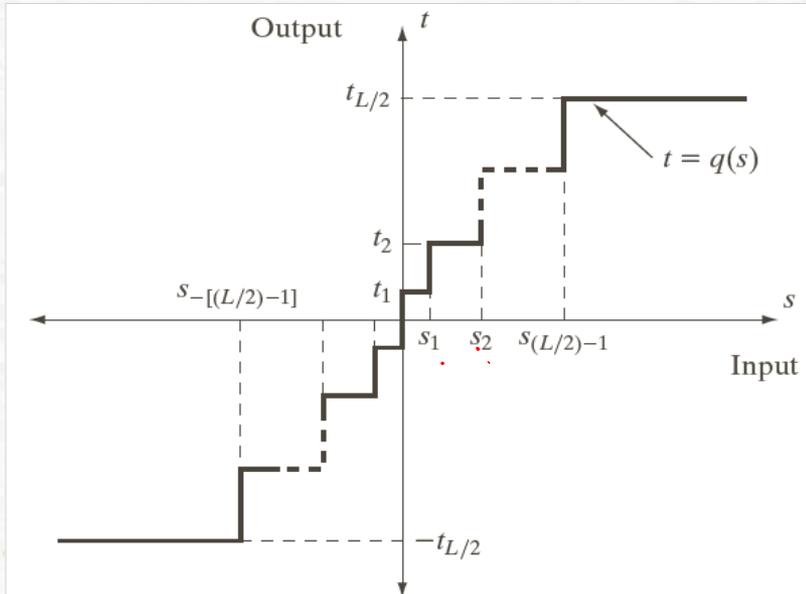
## Exemple de limitations



© 1992-2008 R. C. Gonzalez & R. E. Woods

# Quantification optimale (1)

## Position du problème



© 1992-2008 R. C. Gonzalez & R. E. Woods

À  $L$  donné

- Choix des valeurs de  $s$  ?
- Choix des valeurs de  $t$  ?

Approche : estimation

Minimisation d'une erreur moyenne de quantification ( $E \left[ (s - t(s))^2 \right]$ )

# Plan

---

1. **Codage par transformée par blocs**
  - Démarche générale
  - Choix de la transformée
  - Sélection et quantification des coefficients
  - Redondance structurelle et codage des symboles
  - Exemple : codage JPEG
2. **Codage en ondelettes**
3. **Codage prédictif**
  - Codage prédictif sans et avec perte
  - Structure et spécification des prédicteurs
  - Quantification de l'erreur de prédiction
4. **Éléments de « tatouage numérique » (*watermarking*)**

# Tatouage numérique (*watermarking*)

## Objectifs

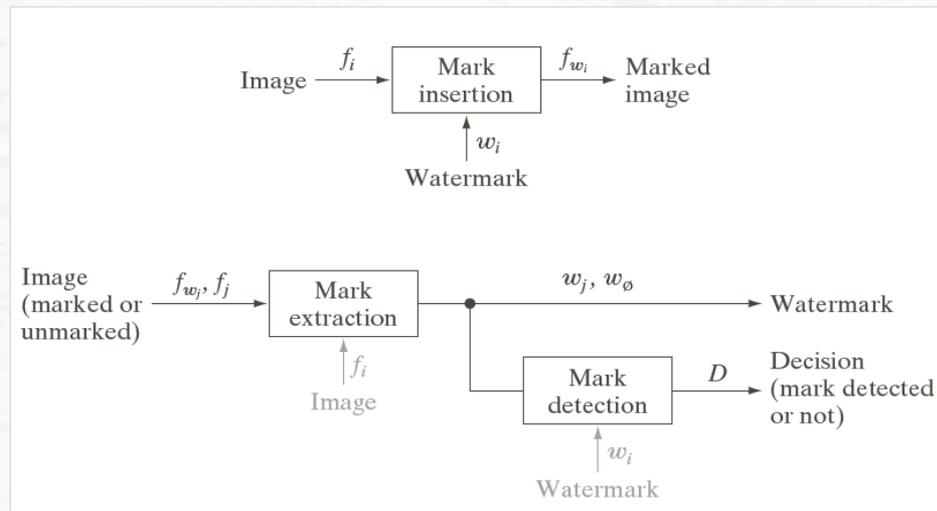
Modification de l'image

- Peu ou pas visible
- Permettant d'authentifier l'origine de l'image
- Résistante ou *non* aux transformations

**Résistant** : résiste aux altérations (compression JPEG, redimensionnement, bruit).

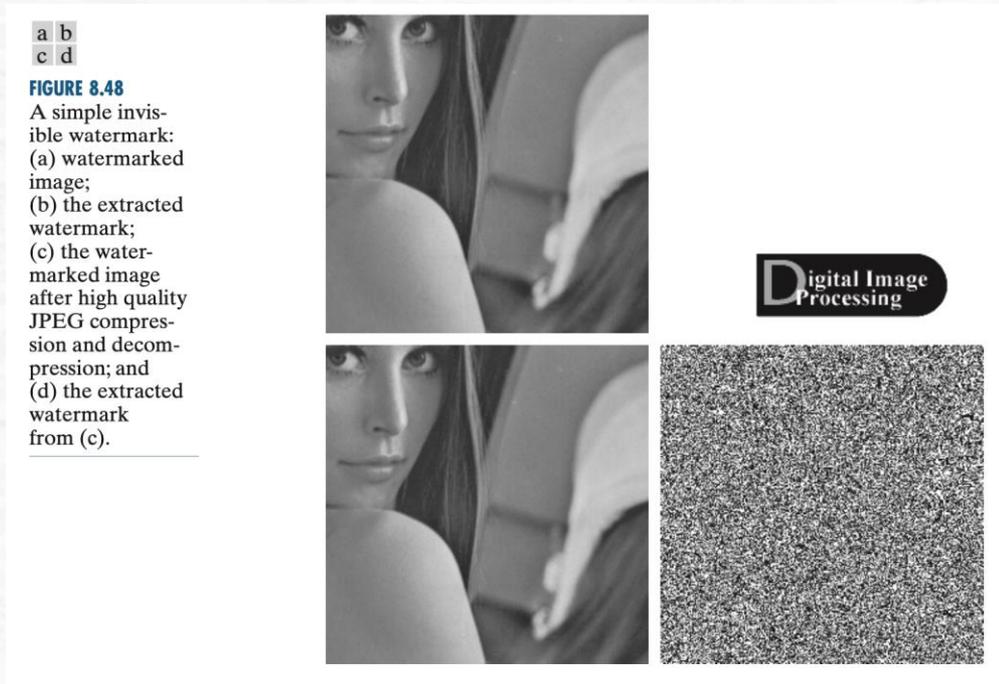
**Fragile** : détecte toute modification → idéal pour valider l'intégrité.

## Structure d'un système de tatouage numérique



# Tatouage non résistant aux transformations

## Exemple : Codage sur les bits de faible poids



- Non perceptible selon son emplacement
- Détruit par la plupart des transformations (transformations géométriques, compression avec pertes,...)

© 1992-2008 R. C. Gonzalez & R. E. Woods

# Tatouage résistant aux transformations

## Exemple : tatouage de la TCD

- Étapes pour le tatouage:
  - calcul de la TCD de l'image et sélection des  $K$  plus grands coefficients
  - génération d'une séquence  $\{\omega_i\}$  de  $K$  nombres pseudo-aléatoires
  - ajout pondéré des  $\{\omega_i\}$  aux coefficients de la TCD
  - Inverse TCD



© 1992-2008 R. C. Gonzalez & R. E. Woods

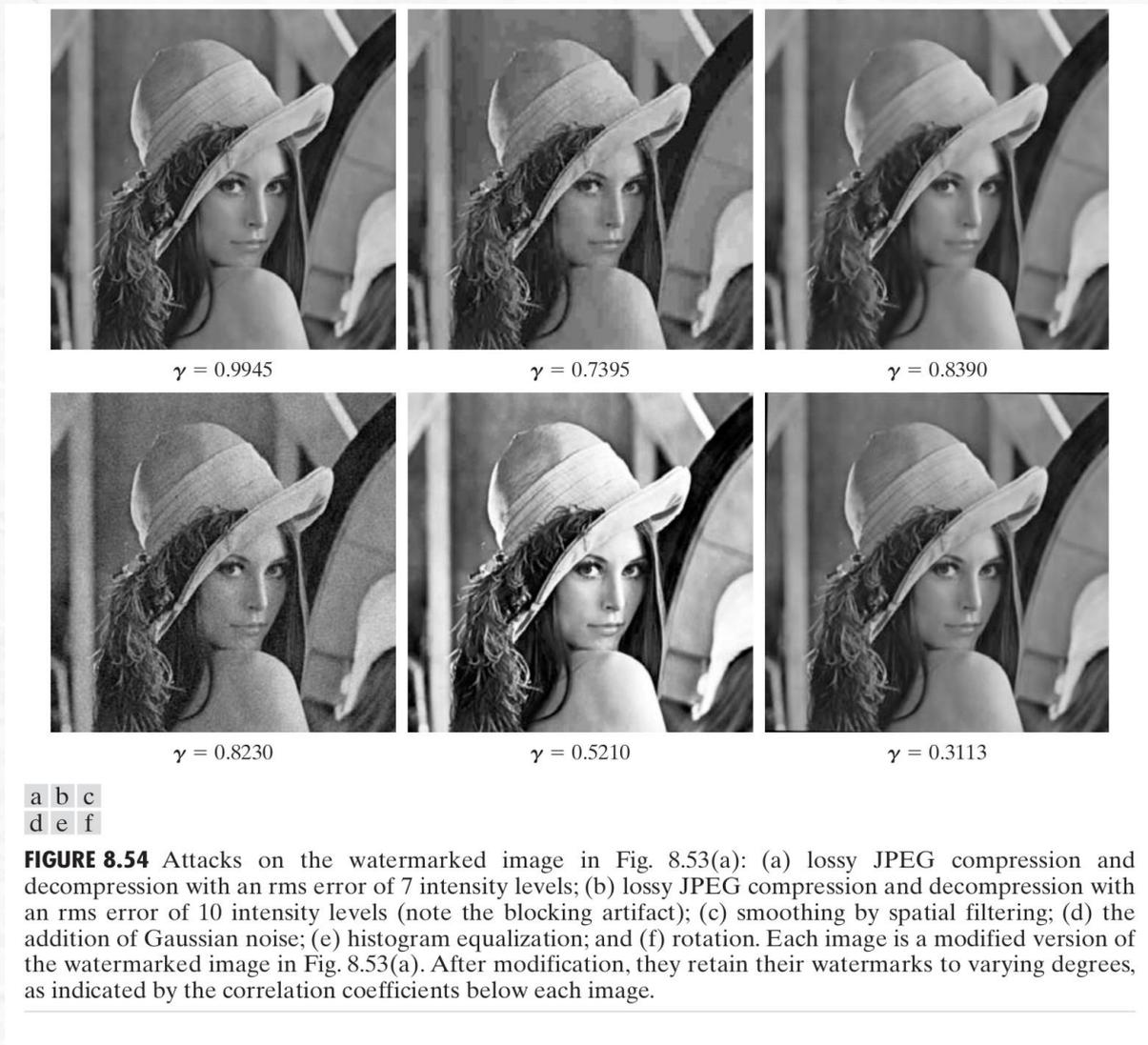
# Tatouage résistant aux transformations

---

## Exemple : tatouage de la TCD

- Étapes pour vérifier si l'image à été modifié:
  - calcul de la TCD de l'image tatouée
  - sélection des coefficients tatoués
  - calcul des  $\{\hat{\omega}_i\}$ , estimée de la séquence pseudo aléatoire
  - décision sur une base de distance entre  $\{\hat{\omega}_i\}$  et  $\{\omega_i\}$

# Tatouage résistant aux transformations



$\gamma$ : coefficient de corrélation entre  $\hat{\omega}_i$  et  $\omega_i$