Compression et codage I

Eva Alonso Ortiz

ELE8812

20 mars 2025

Plan

1. **Notions fondamentales**
   - Introduction
   - Notions de redondance
   - Modèle de codage et normes
   - Notion d’information

2. **Codage de source**
   - Codage de Huffman
   - Codage de Golomb
   - Codage arithmétique
   - Codage de Lempel-Ziv-Welch

3. **Redondance spatiale**
   - Codage par longueur de plage (RLC)
   - Codage par plans de bits (BPC)
   - Codage par symboles

Plan

1. **Notions fondamentales**
   - Introduction
   - Notions de redondance
   - Modèle de codage et normes
   - Notion d'information

2. **Codage de source**
   - Codage de Huffman
   - Codage de Golomb
   - Codage arithmétique
   - Codage de Lempel-Ziv-Welch

3. **Redondance spatiale**
   - Codage par longueur de plage (RLC)
   - Codage par plans de bits (BPC)
   - Codage par symboles

Introduction

Nécessité de compresser les images

- Prévalence des images ou des séquences d'images (loisirs, industrie, santé, …)
- Problème de taille : 1 film de 2h00, basse resolution = 224 Go

Compression nécessaire pour

- Stocker les images
- Transmettre les images (par internet)

Exemples

- Dossier medical informatisé (PACS)
- Archives historiques
- Blogs, Facebook, …
- Photographie numérique

Notions fondamentales

Définitions

- ”Compression de données” : réduire les données utilisées pour transmettre la même information
- Données : les moyens utilisés pour transmettre de l'information

Image originale : b bits ; image compressée : b' bits

Taux de compression : $C = b/b'$

Redondance relative : $R = 1 - 1/C$

Notions fondamentales

Définitions

- Code : système de symboles (lettres, nombres, bits, etc.)
- Chaque élément d'information est associé avec une suite de symboles de code (« mot code »)
- Longueur associée avec chaque mot code

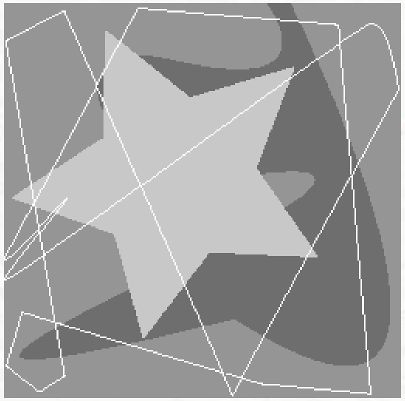
Imagerie

- Utilisation de codes de 8-bits pour représenter des intensités (0-255)
- Pas optimal (plusieurs types de redondances, ex. : zones uniformes codées de manière répétitive)

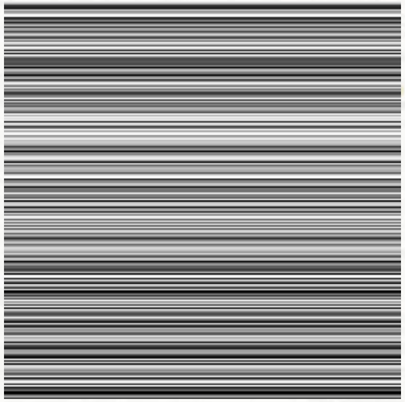
Notions fondamentales

Types de redondance

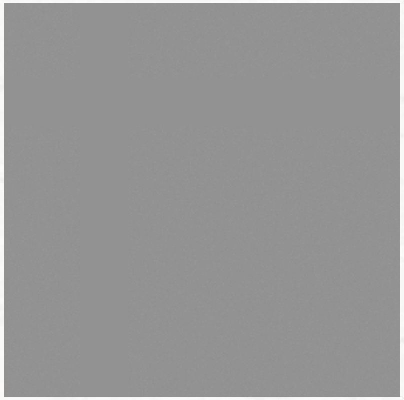
Redondance de source
(4 niveaux de gris, 8 bits)



Redondance spatiale



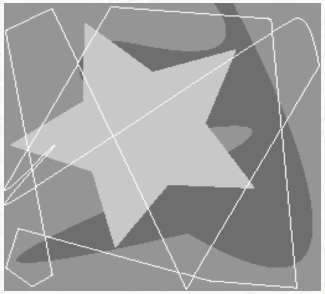
Information non pertinente



Redondance de source

Principe

- Image : suite de symboles *indépendants*
- Exemple de codage à longueur variable :



r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

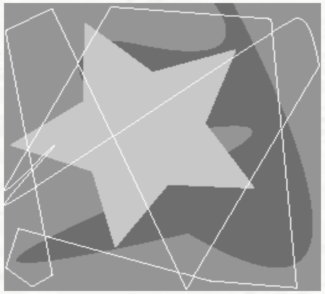
probabilité d'apparition du symbole/intensité $rk = p_r\,(r_k) = \frac{n_k}{MN}$

- $\bar{L} = \sum_k l(r_k)p(r_k)$ $= 0.25 \times 2 + 0.47 \times 1 + 0.25 \times 3 + 0.03 \times 3$
- $\bar{L} = 1,81$ bits =longueur moyenne de bits utilisées

Redondance de source

Principe

- Image : suite de symboles *indépendants*
- Exemple de codage à longueur variable :



r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

- Taille : 256 x 256
- C = ?
- Pourcentage des données qui était redondantes?

slido

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

Slide #8: C=?

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

slido

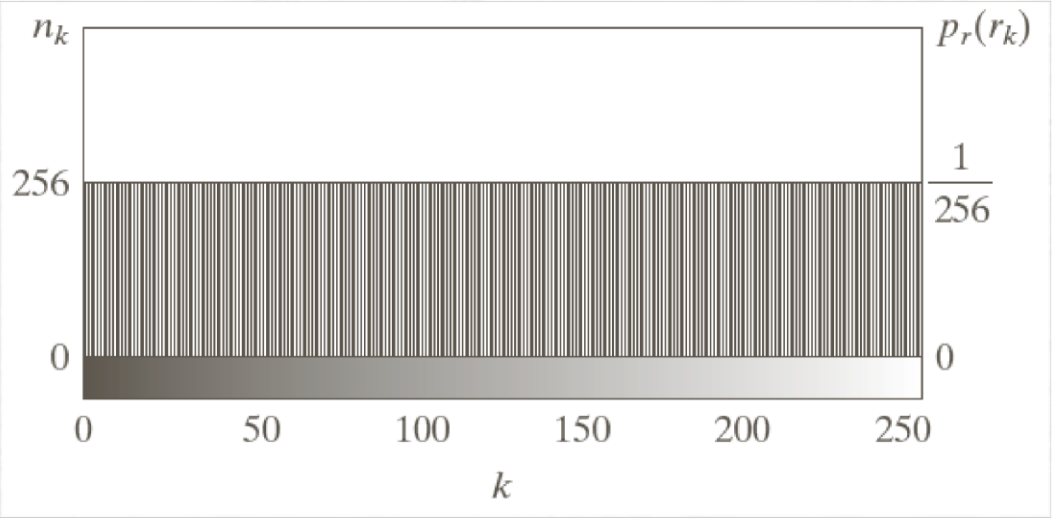
Please download and install the Slido app on all computers you use

Slide #8: pourcentage des données redondantes?

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

Redondance spatiale

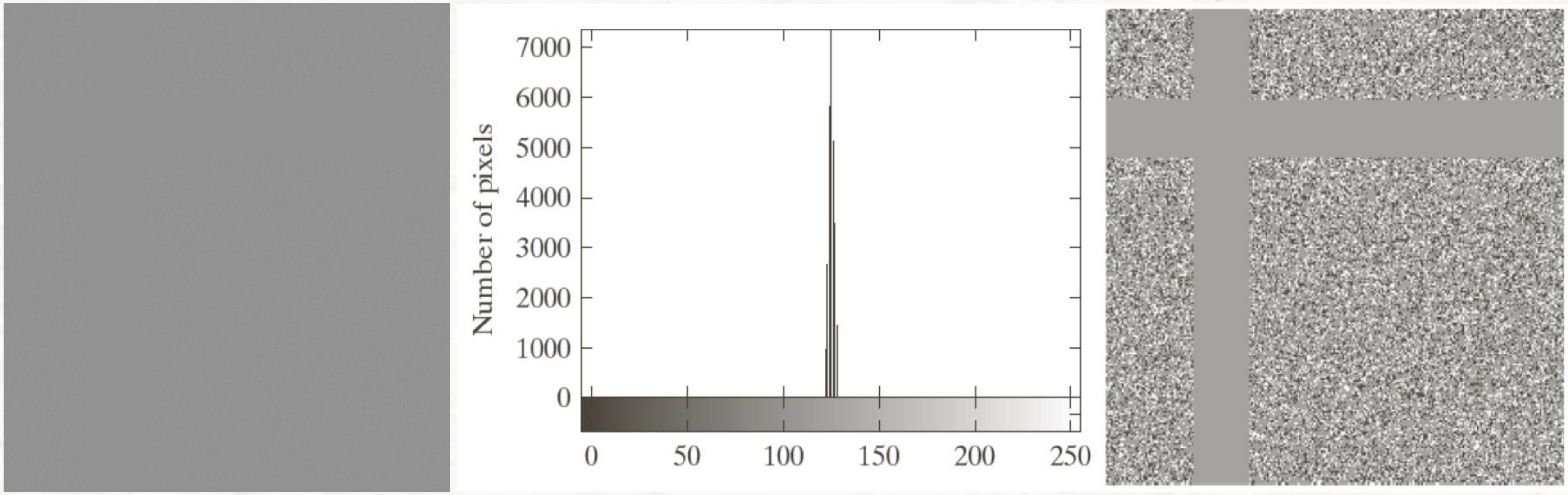
Exemple : codage par longueur de plage (RLC)



- Codage de la longeur et de la valeur de chaque segment horizontal d'intensité constante
- 512 octets : $C = 128$!

Information non pertinente

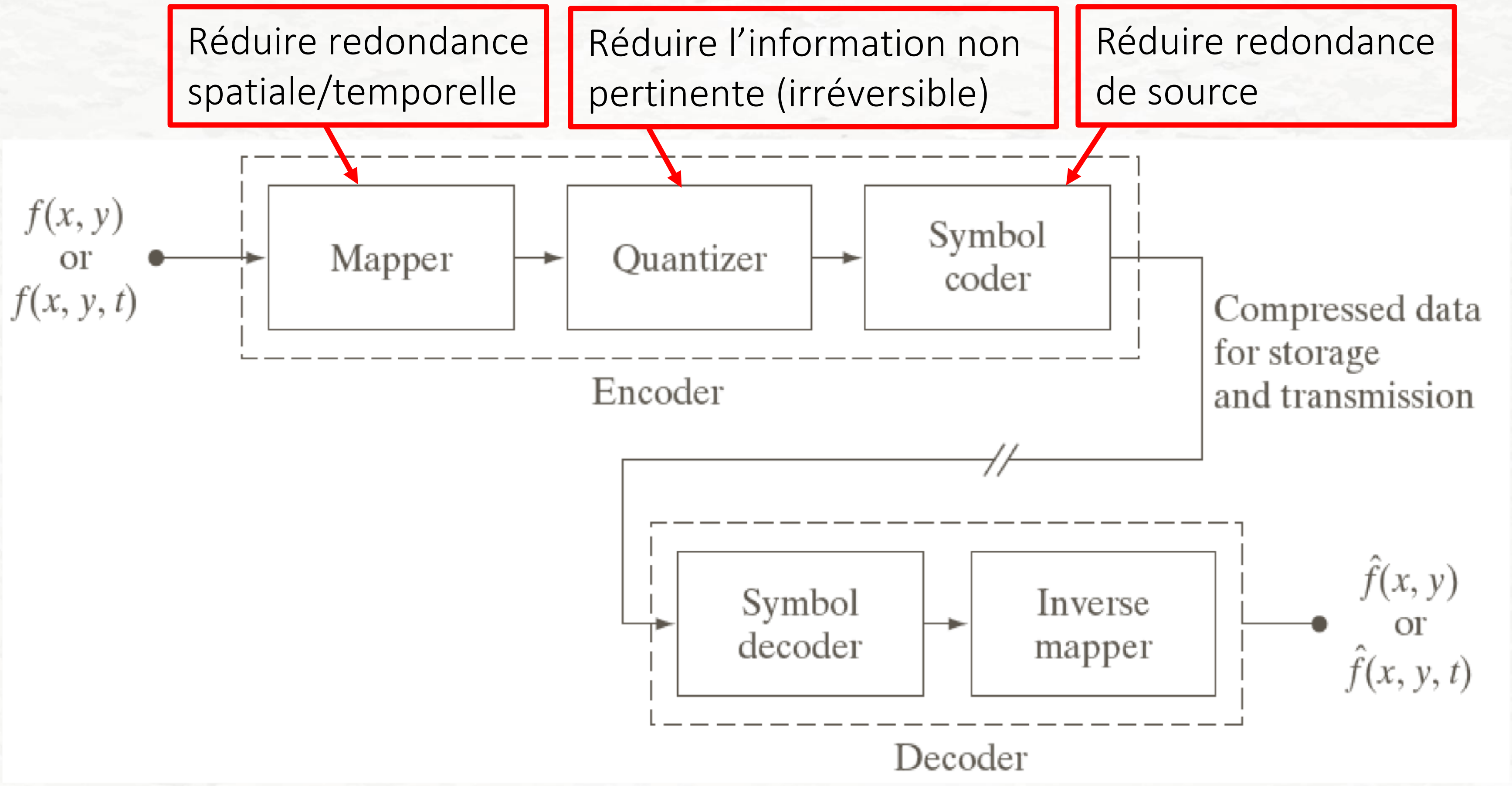
Exemple : image quasi-homogène



- Compression avec perte : une seule valeur. $C = 256^2$!
- Compression sans perte : redondance spatiale *et* redondance de source

Structure d'un système de codage

Principales étapes



Normes, conteneurs et formats (1)

Format de fichier

- Comment les données sont organisées
- Type de compression

GIF	CompuServe	*Graphic Interchange Format*. A file format that uses lossless LZW coding [8.2.4] for 1- through 8-bit images. It is frequently used to make small animations and short low resolution films for the World Wide Web.

Un seul type de données

Conteneur

- Similaire a un format de ficher
- Peut contenir des multiples types de données (images, audio, sous-titres, etc.) ex: PDF, mp4

PDF	Adobe Systems	*Portable Document Format*. A format for representing 2-D documents in a device and resolution independent way. It can function as a container for JPEG, JPEG 2000, CCITT, and other compressed images. Some PDF versions have become ISO standards.

Plusieurs types de données

Normes, conteneurs et formats (2)

Norme de compression

- Définition des procédures pour la compression et décompression d'images

Name	Organization	Description
Bi-Level Still Images		
CCITT Group 3	ITU-T	Designed as a facsimile (FAX) method for transmitting binary documents over telephone lines. Supports 1-D and 2-D run-length [8.2.5] and Huffman [8.2.1] coding.
CCITT Group 4	ITU-T	A simplified and streamlined version of the CCITT Group 3 standard supporting 2-D run-length coding only.
JBIG *or* JBIG1	ISO/IEC/ ITU-T	A *Joint Bi-level Image Experts Group* standard for progressive, lossless compression of bi-level images. Continuous-tone images of up to 6 bits/pixel can be coded on a bit-plane basis [8.2.7]. Context sensitive arithmetic coding [8.2.3] is used and an initial low resolution version of the image can be gradually enhanced with additional compressed data.
JBIG2	ISO/IEC/ ITU-T	A follow-on to JBIG1 for bi-level images in desktop, Internet, and FAX applications. The compression method used is content based, with dictionary based methods [8.2.6] for text and halftone regions, and Huffman [8.2.1] or arithmetic coding [8.2.3] for other image content. It can be lossy or lossless.
Continuous-Tone Still Images		
JPEG	ISO/IEC/ ITU-T	A *Joint Photographic Experts Group* standard for images of photographic quality. Its lossy *baseline coding system* (most commonly implemented) uses quantized discrete cosine transforms (DCT) on 8×8 image blocks [8.2.8], Huffman [8.2.1], and run-length [8.2.5] coding. It is one of the most popular methods for compressing images on the Internet.
JPEG-LS	ISO/IEC/ ITU-T	A lossless to near-lossless standard for continuous tone images based on adaptive prediction [8.2.9], context modeling [8.2.3], and Golomb coding [8.2.2].
JPEG-2000	ISO/IEC/ ITU-T	A follow-on to JPEG for increased compression of photographic quality images. Arithmetic coding [8.2.3] and quantized discrete wavelet transforms (DWT) [8.2.10] are used. The compression can be lossy or lossless.

Arrière-plan mathématique

Notion d'information

- Information contenue par un évènement E :

$$I(E) = -\log_m p(E)$$

p(E)= probabilité de l'évènement E

si m = 2, l'unité d'information est le "bit"

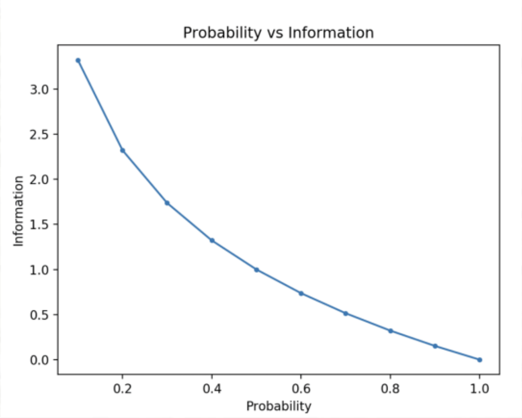
- Ex: Lancer une pièce et communiquer le résultat.

Combien d'information est contenue dans le message qui communique l'évènement en utilisant des bits?

$$p(Face) = p(pile) = 0.5$$

$$I(E) = -\log_2 0.5 = 1\ bit$$

Il faut 1 bit pour coder le résultat de cet évènemnt, 0 pour pile et 1 pour face



https://machinelearningmastery.com/what-is-information-entropy/

Arrière-plan mathématique

Notion d'information

- Information contenue par un évènement E :

$$I(E) = -\log_m p(E)$$

p(E)= probabilité de l'évènement E

si m = 2, l'unité d'information est le "bit"

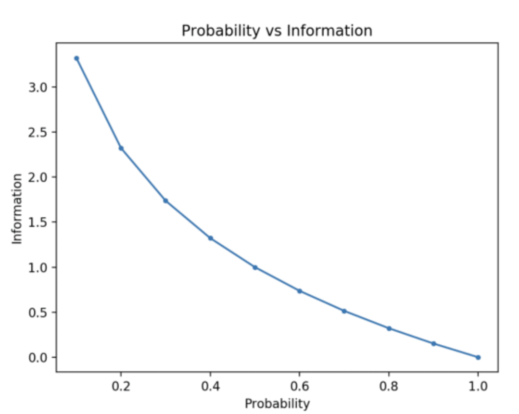
- Ex: Lancer un dé et communiquer le résultat.

Combien d'information est contenue dans le message qui communique l'évènement en utilisant des bits?

$$p(E) = \frac{1}{6} \ \ pour\ tout\ E$$

$$I(E) = -\log_2 1/6 \ \approx 2.58\ bit$$

A mesure que l'événement devient moins probable, plus de bits sont nécessaires



https://machinelearningmastery.com/what-is-information-entropy/

Arrière-plan mathématique

Notion d'information

- Entropie d'une source (symboles/évènements indépendants) :

$$H = \mathrm{E}[I(r)] = -\sum_k p(r_k)\log_2 p(r_k)$$

- Premier théorème de Shannon : H limite inférieure de $\bar{L}$ (longueur moyenne nécessaire pour représenter un pixel ou une intensité)

Attention aux hypothèses sur la source !

- Ex: Selon le théorème de Shannon quel est le nombre minimum de bits/pixel qu'on pourrait utiliser pour encoder l'image d'étoile?

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

slido

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

Selon le théorème de Shannon quel est le nombre minimum de bits/pixel qu’on pourrait utiliser pour encoder l’image d’étoile?

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

Arrière-plan mathématique

Fidélité (codage avec perte)

- $\hat{f}(x,y)$: image résultant du codage de $f(x,y)$
- Mesure la plus courante : $\dfrac{\left\|\hat{f}(x,y)\right\|^2}{\left\|\hat{f}(x,y)-f(x,y)\right\|^2}$

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y)-f(x,y)\right]^2}$$

Mean squared signal to noise ratio

Plan

1. **Notions fondamentales**
   - Introduction
   - Notions de redondance
   - Modèle de codage et normes
   - Notion d'information
2. **Codage de source**
   - Codage de Huffman
   - Codage de Golomb
   - Codage arithmétique
   - Codage de Lempel-Ziv-Welch
3. **Redondance spatiale**
   - Codage par longueur de plage (RLC)
   - Codage par plans de bits (BPC)
   - Codage par symboles

Codage de Huffman (1)

Principe

- Codage de longueur variable (probabilité ↘, longueur ↗)
- Probabilité de chaque symbole connue

Première étape

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Codage de Huffman (2)

Deuxieme étape

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Propriétés

- Quasi-optimalité au sens de la théorie de l'information (si indépendants)

En pratique

- $p(r_k)$? **Indépendance** entre symboles ?
- Quantités à transmettre:
    - symboles et leur probabilité (dictionnaire de Huffman)
    - symboles codés (ton message compressé!)

Codage de Huffman (3)

Ex: Code transmis: 010100111100

Symbol	Probability
a_2	0.4
a_6	0.3
a_1	0.1
a_4	0.1
a_3	0.06
a_5	0.04

Original source		
Symbol	Probability	Code
a_2	0.4	1
a_6	0.3	00
a_1	0.1	011
a_4	0.1	0100
a_3	0.06	01010
a_5	0.04	01011

Symboles décodés?

slido

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

Ex: Code transmis: 010100111100

Codage Huffman: symboles décodés?

Original source		
Symbol	Probability	Code
a_2	0.4	1
a_6	0.3	00
a_1	0.1	011
a_4	0.1	0100
a_3	0.06	01010
a_5	0.04	01011

Start presenting to display the poll results on this slide.

Codage de Golomb (1)

Principe

- Codage de *valeurs entières positives (x)*
- Choix d'un diviseur *m*
- Codage d'une valeur *x* par son quotient *q* et son reste *r*
  - codage *unaire du quotient* $q = \lfloor x/m \rfloor$. *Codage unaire = une série de q fois "1", suivi par un 0*
  - reste : codage de longueur variable (voir P. 613 du livre)

$\lfloor x \rfloor = plus\ grand\ entier\ \leq\ x$
$\lceil x \rceil = plus\ petit\ entier\ \geq x$

Exemple et formules: Valeur = 9, diviseur (m) = 4, $G_m(n) = G_4(9) = ?$

1. Quotient = 2, code unaire = 110 **(le code unaire pour une valeur entière q est une série de q fois "1", suivi par un 0)**

   $q = \lfloor 9/4 \rfloor = \lfloor 2.25 \rfloor = 2$, code unaire: 110

2. On définit $k = \lceil \log_2 m \rceil$ et $c = 2^k - m$ (voir P. 613) pour déterminer le nombre de bits utilisée pour le reste tronqué (*r'*).

$$r' = \begin{cases} r \text{ tronqué a } k-1 \text{ bits} & 0 \leq r < c \\ r + c \text{ tronqué a } k \text{ bits} & \text{sinon} \end{cases}$$

k = $\log_2 4$ = 2, c = 2^2 - 4 = 0, puisque $c < r$, on prend le 2^e cas pour r'

r = 1 + c = **1**, en binaire ça donne 0001, tronqué à k = 2 bits c'est 01

r' = 01

$G_4(9)$ = 11001

Codage de Golomb (2)

n	$G_1(n)$	$G_2(n)$	$G_4(n)$
0	0	00	000
1	10	01	001
2	110	100	010
3	1110	101	011
4	11110	1100	1000
5	111110	1101	1001
6	1111110	11100	1010
7	11111110	11101	1011
8	111111110	111100	11000
9	1111111110	111101	11001

$$P(n) = (1 - \rho)\rho^n$$

$$0 < \rho < 1$$

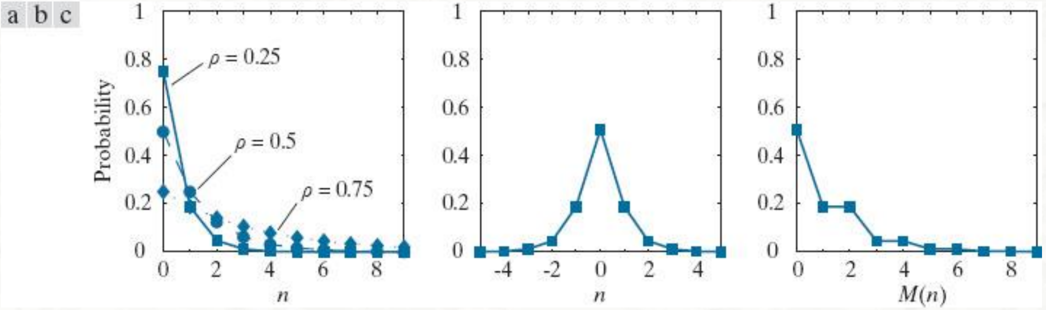
$$m = \left\lceil \frac{\log_2(1 + \rho)}{\log_2(1/\rho)} \right\rceil$$

Formule optimal pour le choix de m

Codage de Golomb (3)

Propriétés

- Quasi-optimalité pour des distributions exponentielles
- Seulement utilisé pour les nombres positifs !
- Adaptation pour les nombres négatifs avec:
- $M(n) = \begin{cases} 2n & n \geq 0 \\ 2|n| - 1 & n < 0 \end{cases}$



En pratique

- Hypothèses sur la distribution des symboles ?
- Quantités à transmettre :
  - valeur de *m*
  - symboles codés

Codage arithmétique (1)

Principe

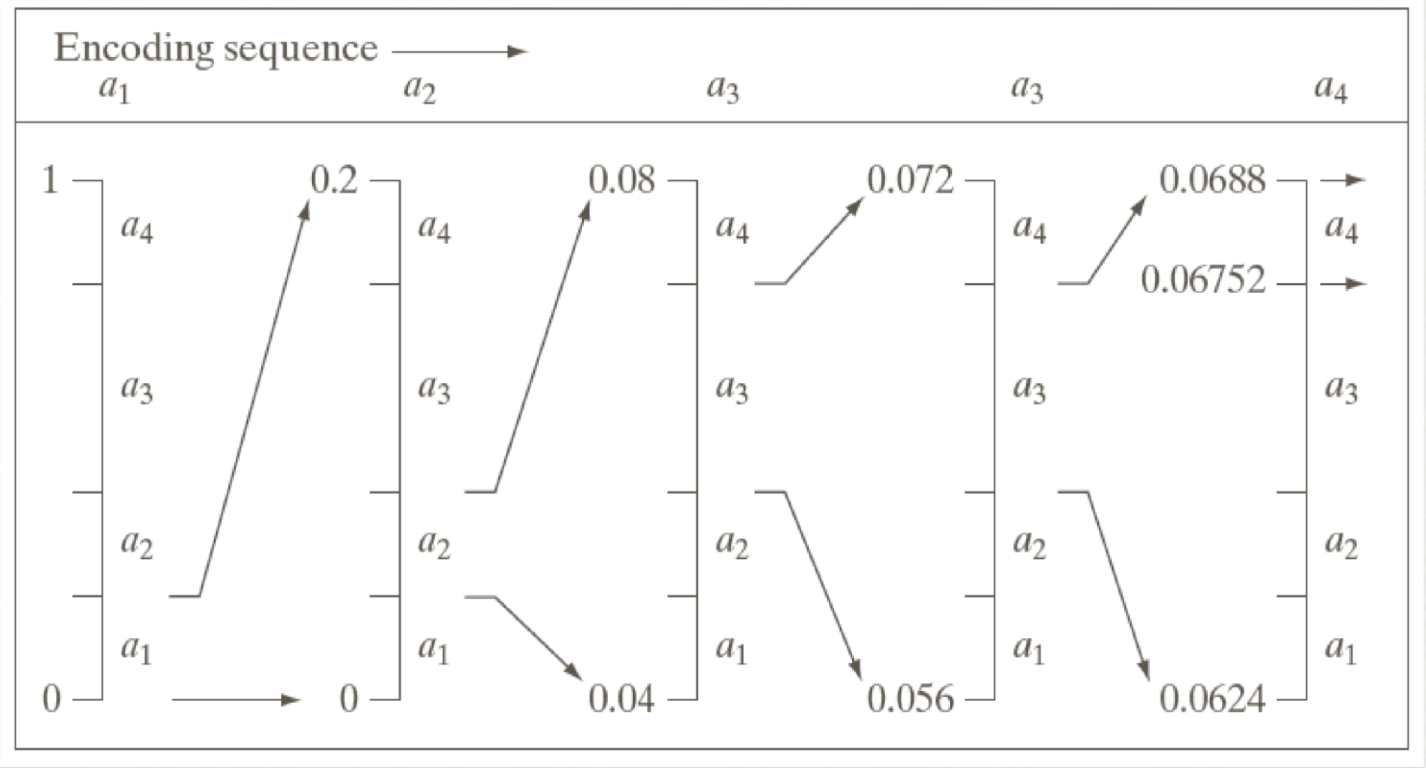
- **Probabilité** de chaque symbole **connue**
- Codage de *suites de symboles* (messages) par un code arithmétique réparti sur une intervalle [0, 1] équivalente

Exemple : $a_1a_2a_3a_3a_4$

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

Codage arithmétique (2)

Exemple : message de 5 symboles



Codage arithmétique (3)

Propriétés

- Quasi-optimal

En pratique

- Nécessité de choisir une représentation arithmétique appropriée de l'intervalle [0, 1]
- Quantités à transmettre :
  - probabilité de chaque symbole
  - code de chaque message
  - code de fin de message

Codage de Lempel-Ziv-Welch (LZW) (1)

Principe

- Ne requière pas la distribution de probabilités des symboles
- Codes de longueur fixe
- Construction récurrente d'un dictionnaire (mot : suite de symboles) au codage *et* au décodage
- Choix de la taille du dictionnaire.
- Pour une image monochrome de 8-bit: Les codes $0-255$ sont réservés pour les intensités $0-255$. Les codes >255 sont assignés a des suites de symboles.

Codage de Lempel-Ziv-Welch (LZW) (2)

Exemple : image (4, 4) sur 256 niveaux

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

TABLE 8.7 LZW coding example.

1. Rentrer pixel actuel
2. **Si** le pixel actuel combiné avec pixels antérieurs correspond a une entrée du dictionnaire
   - i. Passe au prochain pixel et répète étape 1

   Sinon
   - ii. Output encodé: indice du dictionnaire correspondant à la suite actuelle (qui n'inclut pas le pixel actuel)
   - iii. Création d'une nouvelle entrée dans le dictionnaire en combinant la suite actuelle avec le pixel actuel
   - iv. Passe au prochain pixel et répète étape 1.

Codage de Lempel-Ziv-Welch (LZW) (3)

Exemple : image (4, 4) sur 256 niveaux

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

4 x 4 x 8 bits = 128 bits

vs.

10 x 9 bits = 90 bits

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

TABLE 8.7 LZW coding example.

En pratique

Quantités à transmettre :

- Liste de symboles
- Taille du dictionnaire
- Technique de gestion des débordements

[redacted]

Méthode	Idéal pour	Points forts	Limites
Huffman	Données avec symboles fréquents biens définis	Simple, optimal si les probabilités sont bien définies	Nécessite les probabilités
Golomb	Données avec structures géométriques (ex. images)	Très efficace si les données suivent une distribution géométrique	Moins efficace si les données sont variées
Arithmétique	Compression maximale pour toute distribution	Plus proche de l'entropie de Shannon, efficace	Complexité élevée, sensible aux erreurs d'arrondi
LZW	Texte, images, données répétitives	Auto-adaptatif, pas besoin de probabilités	Moins efficace si peu de répétitions

Plan

1. **Notions fondamentales**
   - Introduction
   - Notions de redondance
   - Modèle de codage et normes
   - Notion d'information

2. **Codage de source**
   - Codage de Huffman
   - Codage de Golomb
   - Codage arithmétique
   - Codage de Lempel-Ziv-Welch

3. **Redondance spatiale**
   - Codage par longueur de plage (RLC)
   - Codage par plans de bits (BPC)
   - Codage par symboles

Codage par longueur de plage (RLC) (1)

Principe

- Utilisation de la redondance spatiale
- Description d'une image par :
  - la longueur d'un segment d'intensité constante dans une direction de balayage de l'image
  - la valeur de l'intensité
- Très efficace pour les images binaires (norme en télécopie)

Exemple : format BMP

- Mode codé : 2 octets (longueur du segment, valeur de l'intensité)
- Mode absolu : (premier octet = 0)

Second Byte Value	Condition
0	End of line
1	End of image
2	Move to a new position
3–255	Specify pixels individually

Codage par longueur de plage (RLC) (2)

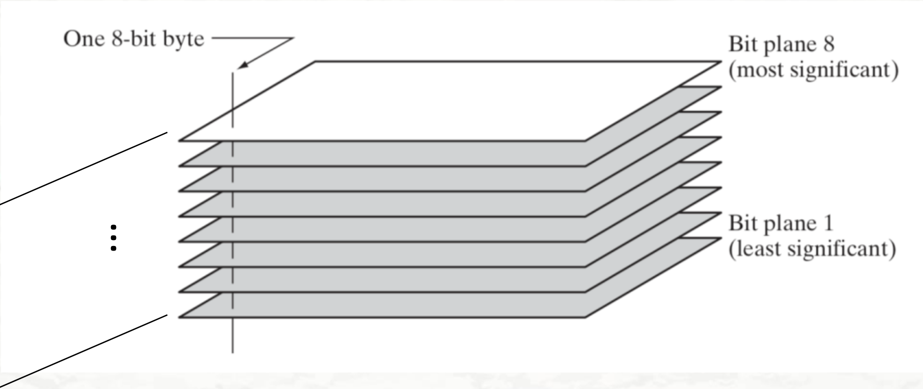
En pratique

- Relativement peu efficace pour les images non binaires
- Peut conduire à un accroissement de la taille des images !
- Peut être associé à un codage de source
- Utilisé dans les normes CCITT

Codage par plans de bits (BPC) (1)

Principe

- Décomposition de chaque pixel de l'image en *k* bits
- Constitution et codage des *k* images binaires



Compression d'image binaire

Compression d'image binaire

Intensité 127 → 01111111
Intensité 128 → 10000000
Tous les images auront une transition 0/1 – 1/0 !

Difficulté : sensibilité à de petites variations d'intensité

Codage par plans de bits (BPC) (2)

Autre possibilité : décomposition en codes de Gray

- $b_{k-1}, b_{k-2}, \ldots, b_0$ valeur des k bits
- Codes de Gray :

$$g_{k-1} = b_{k-1}$$
$$g_i = b_{i+1} \text{ XOR } b_i \quad 0 \leq i \leq k-2$$

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

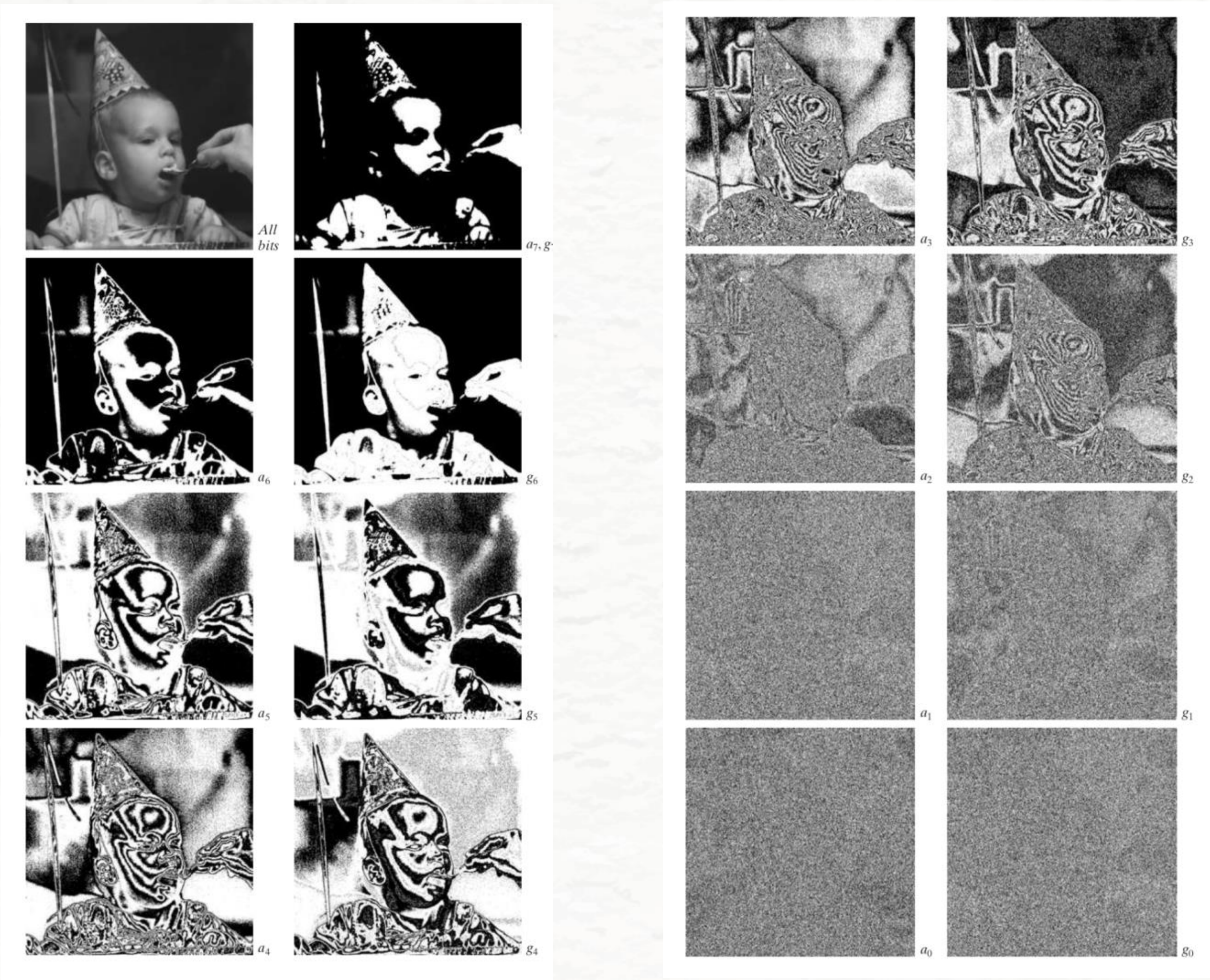
Intensité 127 → 01111111
Intensité 128 → 10000000

Intensité 127 → $\boxed{0}$1000000
Intensité 128 → $\boxed{1}$1000000

Moins de transitions → codage plus efficace!

Codage par plans de bits (BPC) (3)

Décomposition BPC et codes de Gray

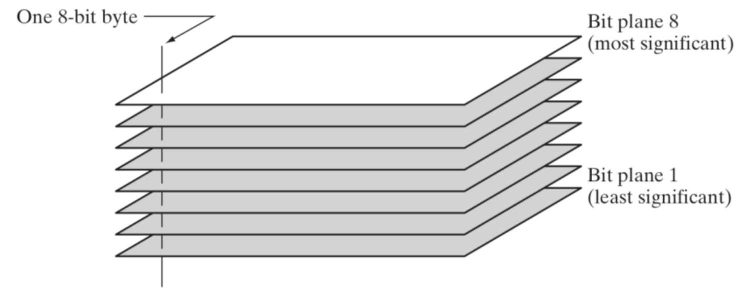


Codage par plans de bits (BPC) (4)

Décomposition BPC et codes de Gray

Codage JBIG2

Coefficient m	Binary Code (PDF bits)	Gray Code (PDF bits)	Compression Ratio
7	6,999	6,999	1.00
6	12,791	11,024	1.16
5	40,104	36,914	1.09
4	55,911	47,415	1.18
3	78,915	67,787	1.16
2	101,535	92,630	1.10
1	107,909	105,286	1.03
0	99,753	107,909	0.92

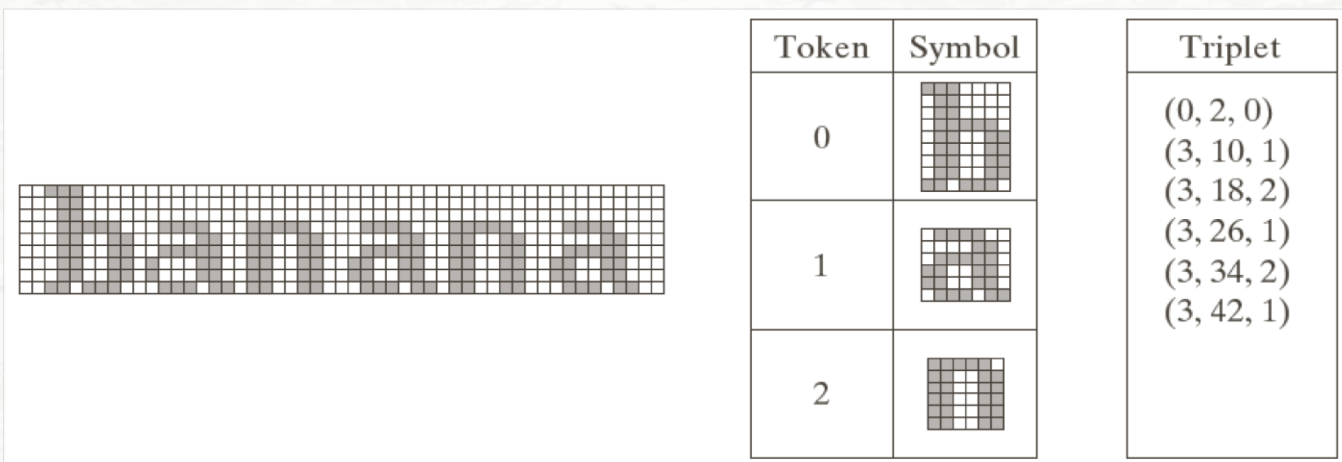


Codage par symboles (1)

Principe

- Reconnaissance de symboles (caractères)
- Création d'un dictionnaire de symboles
- Description de l'image comme un assemblage de ces symboles

Exemple élémentaire

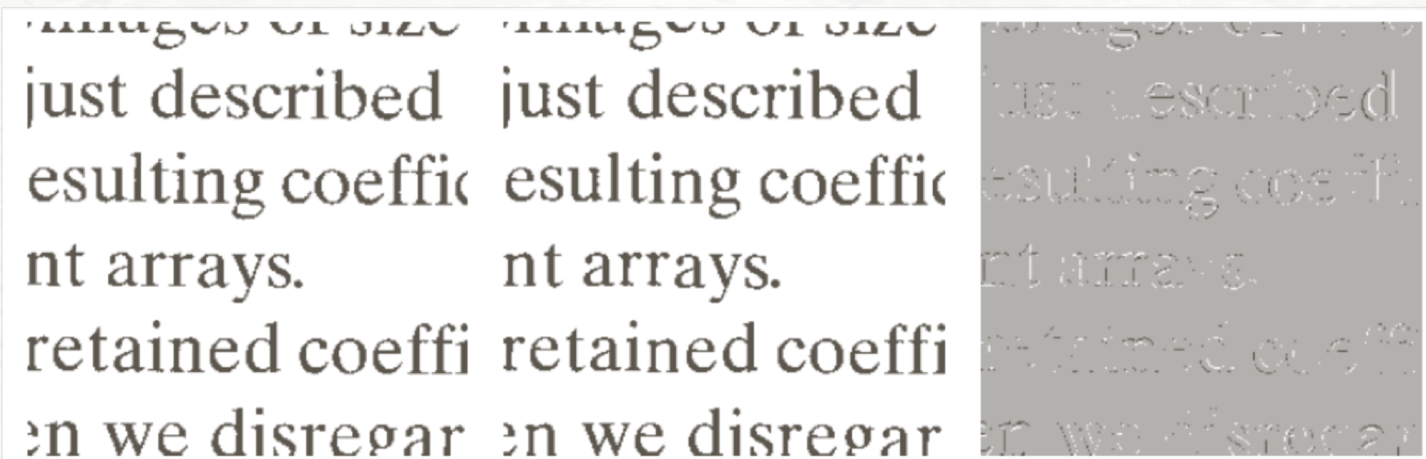


Codage par symboles (2)

Exemple d’utilisation : codage JBIG2

- Segmentation de l’image
- Caractères ou *halftones (motif régulier)* : symboles
- Autres régions (e.g., lignes, bruit) : codage de source

Exemple : distorsion induite par le codage



[redacted]

Critère	RLC	BPC	Codage par symbole
Efficacité	Bonne pour images binaires avec grandes zones uniformes.	Très efficace si combiné avec Huffman ou Arithmétique.	Ultra-efficace pour textes et motifs répétitifs.
Compression avec perte?	Non (mais inefficace si l'image est trop détaillée).	Peut ignorer certains plans de bits pour une compression avec perte.	Peut remplacer des caractères similaires (distorsion possible).
Complexité	Faible (facile à implémenter).	Moyenne (nécessite une analyse des bits).	Élevée (nécessite une reconnaissance et un dictionnaire dynamique).
Robustesse face au bruit	Sensible aux variations de pixels.	Problèmes avec les bits de poids faible.	Bonne, sauf si mauvaise reconnaissance des symboles.
Types d'images idéales	Fax, télécopies, dessins binaire	Images compressées hiérarchiquement, documents scannés.	Texte et documents scannés (OCR, JBIG2).

Exercice

Considérez l'image suivante, de taille 4x8 pixels, codée sur 8 bits (256 niveaux d'intensité) :

7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24

a) Calculez l'entropie de l'image.
b) Calculez le taux de compression obtenu en comprimant l'image uniquement par codage de Huffman.
c) Considérant les caractéristiques de l'image, quelle autre méthode de codage suggéreriez-vous ?

Exercice

Considérez l'image suivante, de taille 4x8 pixels, codée sur 8 bits (256 niveaux d'intensité) :

7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24

a) Calculez l'entropie de l'image

r_k	n_k	$p_r(r_k)$
7	12	12/32
80	12	12/32
13	4	4/32
24	4	4/32

$$H = -\sum_{k=0}^{255} pr(rk) log2\, pr(rk) = 2 * [\frac{12}{32}\, log2(\frac{12}{32}) + \frac{4}{32} log2(\frac{4}{32})] = 1.81 \text{ bits/pixel}$$

Exercice

Considérez l'image suivante, de taille 4x8 pixels, codée sur 8 bits (256 niveaux d'intensité) :

7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24

b) Calculez le taux de compression obtenu en comprimant l'image uniquement par codage de Huffman.

Il y a 4 symboles. Dans le codage de Huffman, le plus probable se verra assigner un code de longueur 1, le second de longueur 2, et les deux moins probables se verront assigner un code de longueur 3. Le nombre moyen de bits nécessaires pour représenter chaque pixel est donc (équation 8.1-4) :

$$L_{avg} = \sum_{k=0}^{255} l(r_k)p_r(r_k) = 1 \times \frac{12}{32} + 2 \times \frac{12}{32} + 3 \times \frac{4}{32} + 3 \times \frac{4}{32} = 1.875\ bits/pixel$$

Le taux de compression, par rapport aux 8 bits nécessaires par pixel dans l'image non-compressée, est donc de $\frac{4\times8\times8}{4\times8\times1.875} = 4.27$.

Exercice

Considérez l'image suivante, de taille 4x8 pixels, codée sur 8 bits (256 niveaux d'intensité) :

7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24
7	7	7	80	80	80	13	24

c) Considérant les caractéristiques de l'image, quelle autre méthode de codage suggéreriez-vous ?

Comme l'image comporte de longues plages de nombres identiques (redondance spatiale), on pourrait utiliser un codage par longueur de plage (run-length coding).