



Université Abderrahmane Mira-Bejaia
Faculté de Technologie

Département Automatique, Télécommunication et Electronique

Laboratoire LTII

Polycopié pédagogique

Kasmi Reda

Titre

CODAGE ET COMPRESSION

Cours destiné aux étudiants de
Master 1 : Télécommunication

Année :2022/ 2023

Table des matières

Contents

Liste des figures.....	iv
1. Chapitre 1. Notions fondamentales de codage source et codage canal	6
1.1. Introduction	6
1.2. Codage source	7
1.3. Codage canal.....	8
1.4. Codage conjoint source/canal	8
2. Chapitre 2 : Codages entropiques.....	9
2.1 Introduction et rappels sur la théorie de l'information	9
2.2 Entropie et mesure de l'information.....	9
a) La quantité d'information d'un événement	9
b) Codage.....	11
c) Condition pour qu'un code soit code à préfixe	11
2.3 Codages entropiques.....	12
a) Codage de Huffman	13
b) Codage de Shannon-Fano.....	18
c) Codage arithmétique.....	19
d) Codage LZW	20
2.4 Efficacité d'un code.....	22
2.5 Conclusion	22
3. Chapitre 3 : Codage Canal.....	23
3.1. Introduction	23
3.2. Schéma simplifié d'un système de communication.....	23
3.3. Modélisation d'un canal	24
3.4. Types de canaux et leurs capacités.....	26
3.4.1. Le canal binaire symétrique (BSC pour Binary Symmetric Channel).....	26
a) La capacité d'un canal binaire symétrique (BSC)	27
b) Calcule de C.....	28

3.4.2.	Le Canal Binaire à Effacement BEC.....	30
a)	La capacité d'un Canal Binaire à effacement BEC.....	31
b)	Calcul de C.....	31
3.5.	Codage correcteur d'erreurs	32
3.5.1.	Les codes de répétition.....	32
3.5.2.	Les codes linéaires (Hamming).....	32
3.5.3.	Code linéaire	34
a)	Code (n, k)-code.....	34
b)	Code systématique.....	35
c)	Matrice génératrice d'un code systématique linéaire	35
d)	Encodage	36
e)	Matrice de contrôle H	36
3.6.	Conclusion	37
4.	Chapitre 4 Méthodes de compression avec pertes	38
4.1	Introduction.....	38
4.2	Compression basées sur les transformations	38
4.2.1	Transformation.....	39
4.2.2	Quantification.....	39
4.2.3	Codages entropiques.....	39
4.3	Critère d'évaluation	40
4.3.1	MSE(Mean Square Error)	40
4.3.2	Peak Signal to Noise Ratio (PSNR).....	40
4.3.3	<i>Structural similarity index measure (SSIM)</i>	41
	Conclusion	41
5.	Chapitre 5 Techniques de compression d'images (Cas du JPEG)	42
5.1	Introduction.....	42
5.2	Algorithme JPEG.....	42
5.2.1	Transformation de couleur	43
5.2.2	Sous échantillonnage.....	44
5.2.3	Transformation en Cosinus Discret (DCT).....	45
5.2.4	Quantification	46

5.2.5	Balayage en <i>séquence zigzag</i>	46
5.2.6.	Compression RLE	47
5.3	Compression d'images à base d'ondelette et Algorithme EZW	47
5.3.1	Transformée en ondelette.....	47
5.3.2	Compression d'image avec l'Algorithme EZW	49
5.4	Comparaison de JPEG et EZW.....	51
5.5	Conclusion	52
CONCLUSION		53
REFERENCES		54

Liste des figures

Figure 1.1	Schéma d'un système de transmission	6
Figure 3.1	Un schéma général du système de la communication selon Shannon	23
Figure 3.2	schéma du canal BSC.....	26
Figure 3.3	Information mutuelle entre X et Y	28
Figure 3.4	Erreur de transmission Canal Binaire Symétrique	29
Figure 3.5	Schéma canal à effacement.....	30
Figure 3.6	Capacité d'un canal à effacement en fonction de p.....	32
Figure 4.1	les étapes de compression avec perte d'une image	38
Figure 4.2	Quantification d'un signal 1D	39
Figure 5.1	Algorithme de compression décompression de l'image norme JPEG.....	43
Figure 5.2	Transformation d'image RGB vers YCbCr	43
Figure 5.3	Transformée en cosinus de l'image (a) image originale (b) sa DCT	45
Figure 5.4	Scan en zigzag de l'image	47
Figure 5.5	Décomposition de l'image à multi-résolutions.....	48
Figure 5.6	Décomposition d'une image en ondelette	48
Figure 5.7	Deux différents scans de l'image (a) Scan Rater (b) Scan Morton	50

Introduction Générale

Le codage et la compression sont deux étapes fondamentales dans la chaîne de transmission de l'information. Elles sont utilisées dans le traitement des données pour diverses raisons. Le codage est le processus de conversion d'informations d'un format à un autre. Il est utilisé pour réduire la taille des données, faciliter la transmission et le stockage des données, la sécurisation des données et l'amélioration de l'efficacité du traitement des données.

La compression, quant à elle, est une technique qui a pour but la réduction de la taille des données. La compression est utilisée pour économiser de l'espace de stockage, réduire le temps de transmission des données sur les réseaux.

Le polycopié est destiné aux étudiants de master 1 spécialité systèmes et réseaux des télécommunications.

Pour assimiler les différentes techniques et définitions illustrées dans ce polycopié, il est recommandé aux étudiants d'avoir aux préalables des connaissances sur les probabilités et statistiques, théorie d'information, Traitement du signal.

Le premier chapitre introduit les notions fondamentales du codage de source et de canal.

Le deuxième chapitre de ce polycopié est consacré au codage de source en présentant quelques techniques de codage de source.

Le troisième chapitre présente le codage canal et explique les techniques de codage canal.

Le quatrième chapitre introduit les techniques de compression avec perte en se basant sur les transformations de l'information.

Le cinquième chapitre est consacré à la compression de l'image en prenant le cas JPEG.

Chapitre 1. Notions fondamentales de codage source et codage canal

1.1. Introduction

Les systèmes de communication numériques consistent à transmettre l'information sur un support physique sous forme de signal numérique. Une information émise par une source (générateur de données), subit un certain nombre de modifications et de manipulation avant que le récepteur puisse la restaurer le plus fidèlement possible. La figure 1.1 montre les trois blocs principaux qui schématisent un système de communication avec des perturbations externes (bruits) sur le canal de transmission.

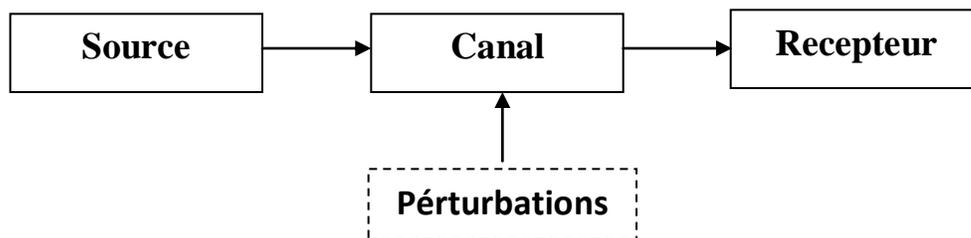


Figure 1.1 Schéma d'un système de transmission

La source : génère les données d'une manière aléatoire ou non aléatoire, elle peut être signal audio, vidéo, image, texte, ...

Canal : c'est le support où l'information est acheminée, elle peut être un fil coaxial, fibre optique, ondes électromagnétiques,....

Récepteur : c'est le destinataire de l'information généré par la source.

Perturbations : l'ensemble des bruits qui agissent et altère l'information au niveau du canal de transmission

Avant de transmettre les informations sur un canal, les données subissent un codage au niveau de la source qu'on appelle codage source et un autre codage au niveau du canal qu'on appelle codage canal.

1.2. Codage source

Est une technique de compression de données qui sert à représenter les données avec un nombre minimum de bits en gardant l'information essentielle donc réduire la longueur du message à transmettre.

Dans les années 1940-1950, Shannon a proposé de compresser des messages numériques en calculant la probabilité d'apparition de chaque message, et les longueurs des codes sont proportionnellement inversées aux probabilités d'apparitions des messages à coder. Ainsi, à partir une description statistique, on peut atteindre une compression maximum sans perte.

Exemple: on veut compresser le mot « Mississippi »

Le codage sans perte :

- la méthode la plus évidente (Trivial) c'est d'associer un mot de code binaire pour chaque lettre. Le mot « Mississippi » est constitué de 4 lettres (M, I, S, P) donc on peut coder chaque lettre avec un code de 2 bits.

$$M = 00, I = 01, S = 10, P = 11$$

Le mot peut être codé sur 22 bits : 00 01 10 10 01 01 01 01 11 11 01

- Une méthode plus efficace tiendra compte de la probabilité de l'apparition de chaque lettre (entropique), ce qui donnera des mots de codes de différentes tailles.

$$M = 001, I = 0, S = 1, P = 01$$

Cela nous permet de coder le mot « Mississippi » sur 15 bits

Le codage avec perte:

On enlève les répétitions des lettres qui constituent le mot *Mississippi*. Le mot devient *Misisipi*, et le codage est :

- **codage « trivial »**

Misisipi → 00 01 10 01 10 01 11 01

Donc sur 16 bits

- **codage « entropique »**

Misisipi → 000 0 1 0 1 0 01 0

Donc sur 11 bits

Remarque: on utilisant le codage avec perte, on ne peut pas retrouver le mot original *Mississippi*

1.3. Codage canal

Le but du codage canal est de détecter d'éventuelles erreurs lors de la transmission des données et de tenter de les corriger. En effet, le codage canal consiste à ajouter de l'information, donc un certain nombre de bits, aux données à transmettre pour ainsi créer de la redondance. Lors du décodage à la réception, la redondance introduite par le codage canal est exploitée pour détecter et corriger des erreurs ayant pu introduites par le canal.

1.4. Codage conjoint source/canal

Le codage source a pour but de compresser, donc de réduire les redondances d'information à transmettre et ainsi réduire la quantité des données. D'autre part, le codage canal a pour but de détecter des erreurs de transmission en ajoutant des données à l'information à transmettre. Selon le théorème de Shannon, les deux fonctions contradictoires du codage source et canal n'affectent pas les performances et l'efficacité du système global. Cependant, ce codage séparé nécessite des codeurs complexes et des délais plus longs. D'où l'introduction du codage conjoint source/canal caractérisé par la simplicité des décodeurs et de ne pas ajouter des données pour détecter les erreurs.

Chapitre 2 : Codages entropiques

2.1 Introduction et rappels sur la théorie de l'information

La théorie de l'information issue des travaux de Claude Elwood Shannon dans les années cinquante représente la base des télécommunications actuelles.

Ce chapitre s'intéresse au codage de source dans la chaîne de transmission de l'information.

- **Une source discrète** : est un générateur aléatoire de symboles issus d'un ensemble fini. Un signal ou une image digitale sont des sources discrètes. L'ensemble des symboles aléatoires de la source est appelé Alphabet noté Ω .
Tel que $\Omega = \{X_0, X_1, \dots, X_N\}$
- **Source sans mémoire** : une source est dite sans mémoire si tous les symboles générés sont indépendants et suivent la même loi de probabilité. Donc la source sans mémoire génère des variables aléatoires identiquement distribuées.

2.2 Entropie et mesure de l'information

Rappelons que l'information est d'autant plus importante lorsqu'elle est improbable et faible quand elle est certaine. Ce qui fait, l'importance de l'information est inversement proportionnelle à la probabilité d'occurrence de l'événement.

a) La quantité d'information d'un événement

La quantité d'informations du message est proportionnelle à son degré d'incertitude. **La quantité d'informations d'un message est une mesure de son imprévisibilité.**

Soit X une variable aléatoire, et que $y \in X$ un événement de X dont la probabilité $p(y)$ est non nulle. **La quantité d'information** apportée par l'événement y est mesurée:

$$I(y) = -\log_2(p(y)) \quad (2.1)$$

L'unité élémentaire de mesure de quantité d'information est le bit.

Supposant maintenant que la variable aléatoire X contient plusieurs événements y_1, y_2, \dots, y_N . La quantité d'information apportée par l'ensemble des événements est calculée par **l'entropie de Shannon**, qui est la **moyenne des informations** apportées par l'ensemble des événements.

L'entropie de Shannon d'une variable aléatoire X est $H(X)$:

$$H(X) = -\sum_{y \in X} p(y) \log_2(p(y)) \quad (2.2)$$

Pour une source sans mémoire, un message de N symboles contient en moyenne une quantité d'information égale à $H(x)N$.

Shannon stipule qu'il est **impossible** de trouver un code dont la **longueur moyenne soit inférieure à $H(x)N$** , mais qu'il est possible de coder le message de façon que le message codé ait en moyenne une longueur aussi proche que l'on veut de $H(x)N$ lorsque N augmente.

Lorsque tous ses symboles sont équiprobables l'entropie $H(X)$ est maximale et vaut $\log_2(n)$ où n est le nombre de symbole de la source.

Exemple : soit une source de 8 symboles équiprobables, la moyenne d'information par symbole est : $\log_2(8)=3$

La redondance d'une source donnée est calculée : $H_{\max}(X) - H(X)$

b) Codage

Pour qu'on puisse décoder un code, il faut qu'il satisfasse la condition à préfixe: aucun code ne doit être préfix d'un autre code. La condition du préfixe assure l'unicité du décodage du code.

Exemple:

{010, 111, 100} : aucun code n'est préfixe de l'autre donc c'est un code à préfixe

{**01**, **011**, 100} : le premier code est préfixe du deuxième code ce n'est pas un code à préfixe.

Problème avec les codes non à préfixe

Soit le mot BEJAIA codé ainsi:

Symbole	Probabilité	Code n'est pas à préfixe
B	1/6	00
E	1/6	01
J	1/6	10
A	2/6	111
I	1/6	11

BEJAIA → 00 01 10 111 11 111

On peut décoder ainsi : **00 01 10 11 111 11 → BEJIAI**

c) Condition pour qu'un code soit code à préfixe

Kraft a défini des conditions pour qu'un code soit uniquement décodable

Théorème 1 (Inégalité de Kraft)

Soient $x \in X$ une source et C un code (mot binaire) uniquement décodable pour X . Alors

$$\sum_{x \in X} 2^{-l(c(x))} \leq 1 \quad (2.3)$$

$l(c(x))$: Longueur du code $C(x)$

Théorème 2

Soit X une source, $x \in X$, $X \sim p$. Alors, il existe un code de préfixe de longueur moyenne :

$$L_C(x) = \sum_{x \in X} p(x) l(c(x)) \quad (2.4)$$

Théorème 3 (Théorème de Shannon)

Soit $x \in X$, $X \sim p$, avec entropie $H(X)$. Alors, il existe un code de préfixe C qui code X avec longueur moyenne à moins d'un bit d'entropie :

$$H(x) \leq L_C(x) < H(x) + 1 \quad (2.5)$$

Théorème 4 (redondance)

La **redondance d'un code** C (préfixe) par rapport à une source $X \sim p$, $x \in X$, est bornée inférieurement par l'entropie relative :

$$R = L_C(X) - H(X) \geq \sum_{x \in X} p(x) \log\left(\frac{p(x)}{q_C(x)}\right) \quad (2.6)$$

Où

$$q_C(x) = \frac{2^{-l(c(x))}}{\sum_{x \in X} 2^{-l(c(x))}} \quad (2.7)$$

2.3 Codages entropiques

Le codage entropique dit statistique à longueur variable est un codage sans perte. Le codage de Huffman, Shannon-Fano et le codage arithmétique sont des codages entropiques.

a) Codage de Huffman

Proposé par David Huffman en 1952, cet algorithme de codage se base sur les probabilités d'apparitions des symboles. Les codes sont à longueurs variables, les codes les plus courts sont attribués aux symboles dont la probabilité d'apparition est élevée.

Le code de Huffman est un *code optimal* : un code préfixe (aucun code n'est préfixe d'un autre code) de longueur moyenne minimale.

Algorithme de Huffman

Le principe est de construire un *arbre binaire* dont les nœuds de départ sont les probabilités des symboles. Puis, d'autres nœuds sont créés en sommant à, chaque fois, les deux nœuds les plus faibles.

1. Calculer les probabilités de chaque symbole à coder
2. Placer les probabilités des symboles par ordre croissant (les nœuds de l'arbre)
3. Construire un nouvel arbre binaire en sommant les deux nœuds les plus faibles, la nouvelle valeur pointe vers les deux nœuds qui l'ont créé
4. Recommencer de 1 jusqu'à ce qu'il reste un seul nœud

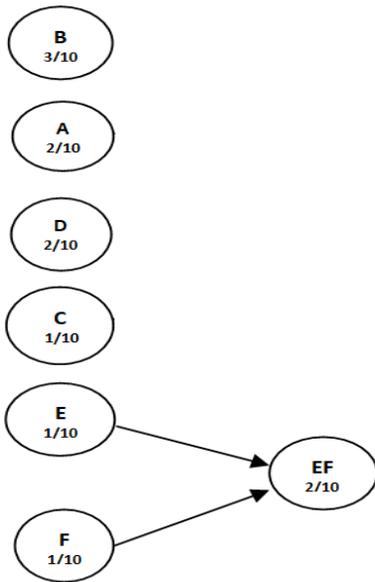
En partant de la racine de l'arbre final, on place pour chaque embranchement un '0' à gauche et '1' à droite.

Exemple: Soit la table de la statistique des symboles A, B, C, D, E, F suivante :

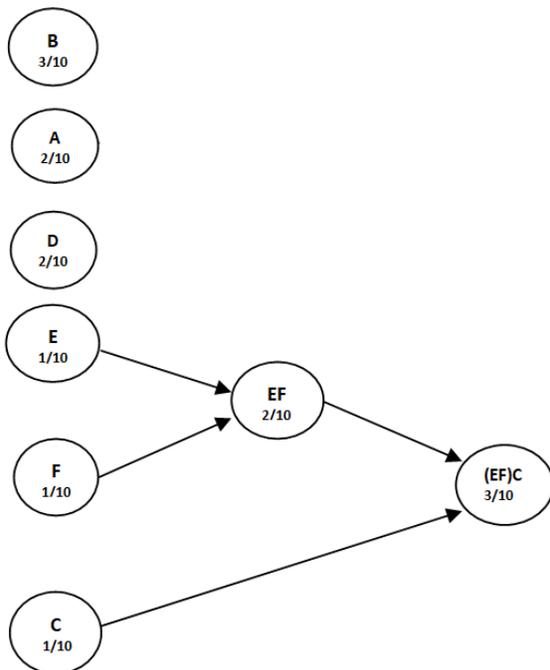
Symboles	Nombre d'apparition	Probabilité
A	2	2/10
B	3	3/10
C	1	1/10
D	2	2/10
E	1	1/10
F	1	1/10

Placer les symboles par ordre croissant, puis construire l'arbre binaire

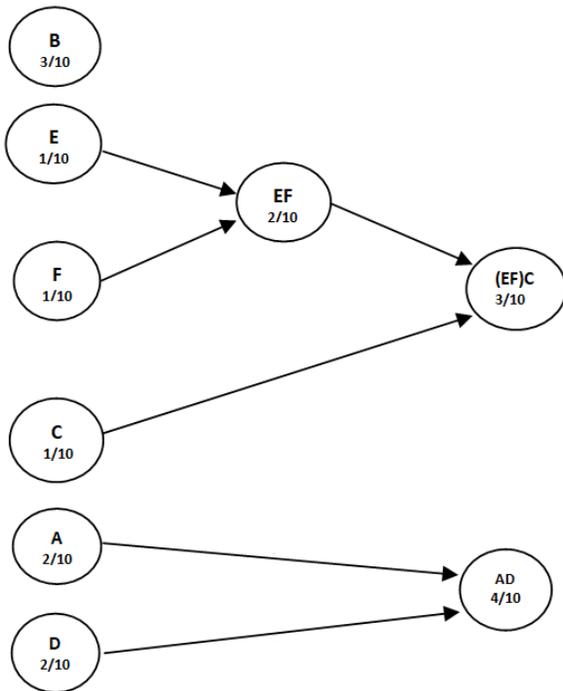
Étape1:



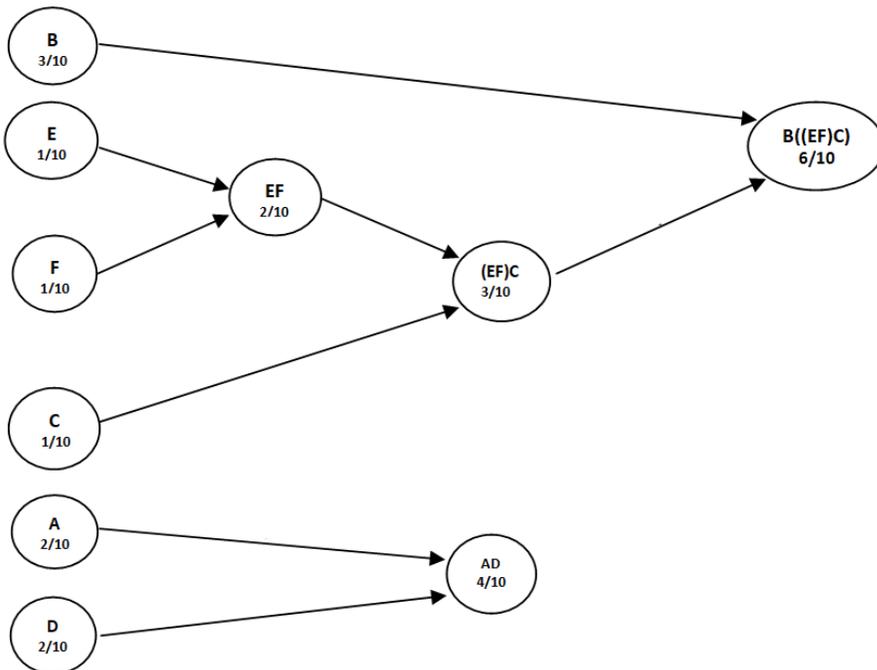
Étape 2 :



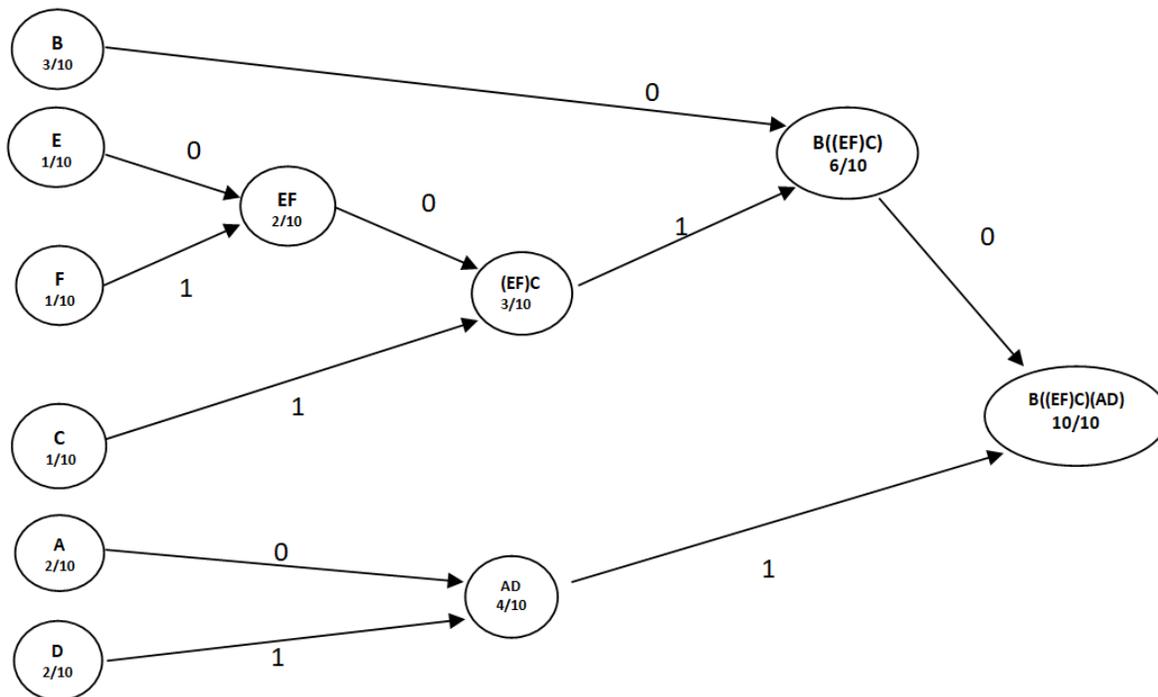
Étape 3 :



Étape 4 :



Étape 5 :



D'où les codes : A : '10' ; B : '00' ; C : '011' ; D : '11' ; E : '0100' ; F : '0101' ;

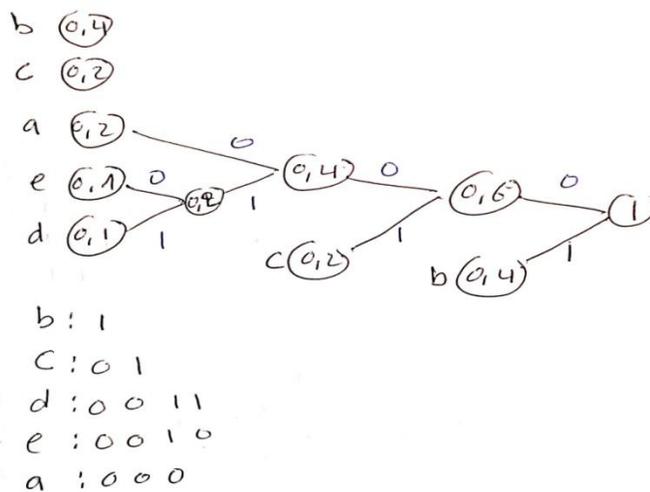
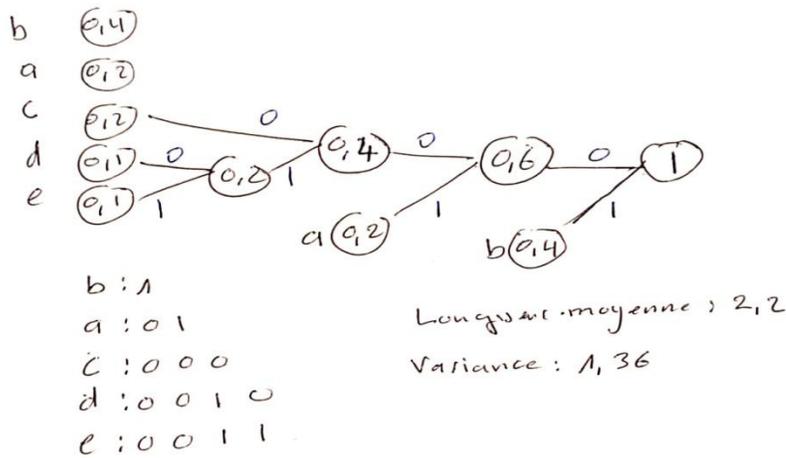
Symboles	Probabilité	Code
B	3/10	00
A	2/10	10
D	2/10	11
C	1/10	011
E	1/10	0100
F	1/10	0101

Remarque

Pour un même ensemble de symboles à coder, plusieurs codes de Huffman différents peuvent être obtenus.

Il est possible de transformer un code de Huffman en un code de Huffman canonique qui est unique pour un ensemble de symboles d'entrée donné. Le principe est d'ordonner au départ les symboles dans l'ordre lexical.

Exemple

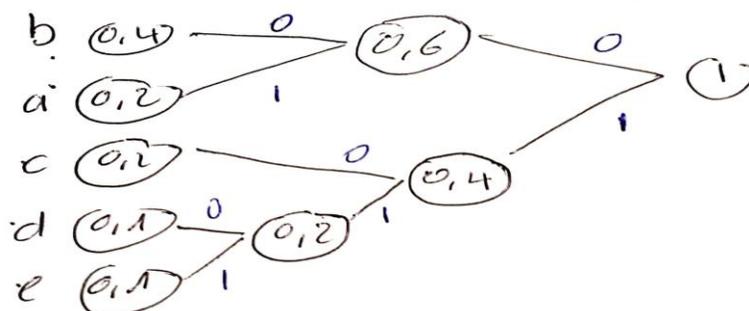


Génération du code de Huffman à variance minimale

- Trouver un code Huffman en minimisant la variance (le code avec une variance minimale prend moins d'espace mémoire).

L'algorithme :

- S'il y a égalité de plus de deux probabilités (plus de deux nœuds équivalent) on choisit les deux nœuds qui donnent la variance globale minimale.
- Dans l'exemple qu'on a présenté avant la variance est de 1.36. On va construire un code de Huffman à variance minimale de 0.24



b : 00
 a : 01
 c : 10
 d : 110
 e : 111

Longueur moyenne : 2,2
 Variance : 0,24

b) Codage de Shannon-Fano

Est un algorithme de codage sans perte, proposé par les deux chercheurs Shannon et Fano (1948/ 1949), d'où le nom de l'algorithme.

1. Trie les probabilités des symboles par ordre croissant
2. Séparé en deux groupes égaux ou presque
3. On réitère l'opération jusqu'à obtenir un seul nœud.
4. Finalement on associe pour chaque sous-groupe le code 0 au premier groupe et 1 au deuxième groupe.

xi	P(xi)	Etape 1	Etape 2	Etape 3	Etape 4	Code
B	3/10	0	0			00
A	2/10	0	1			01
D	2/10	1	0			10
C	1/10	1	1	0		110
E	1/10	1	1	1	0	1110
F	1/10	1	1	1	1	1111

c) Codage arithmétique

Contrairement au codage de Huffman, le codeur arithmétique traite un bloc de symbole en lui associant un unique nombre décimal rationnel.

Algorithme

1. Trier les symboles du bloc à coder selon leur probabilité d'apparition
2. Définir les intervalles pour chaque symbole [Borne-Inférieur Bonne-Supérieure]
 - Borne inférieure (BI) commence de 0 pour le premier symbole. Et la BI pour le symbole (n) correspond à la borne supérieure (BS) du symbole (n-1)
 - Les BS correspondent aux probabilités d'apparition des symboles
3. Le codage s'opère progressivement du premier symbole selon les opérations suivantes :

$$BI(n) = BI(n-1) + (BS(n-1) - BI(n-1)) \times BI_du_symbole$$

$$BS(n) = BI(n-1) + (BS(n-1) - BI(n-1)) \times BS_du_symbole$$

Exemple: codage arithmétique du mot « Bejaia »

Symboles	Probabilité	Intervalles
B	1/6 = 0.166	[0 0.166[
E	1/6	[0.166 0.333[
J	1/6	[0.333 0,499[
I	1/6	[0,499 0,665[
A	2/6	[0,665 1[

Le codage :

Symboles	BI	BS
B	0	0.166
E	$0+(0.166-0)\times 0.166=0.027$	$0+(0.166-0)\times 0.333=0.055$
J	$0.027+(0.055-0.027)\times 0.333 =0.036$	$0.027+(0.055-0.027)\times 0.499=0.04$
A	$0.036+(0.04-0.036)\times 0,665 =0.0386$	$0.036+(0.04-0.036)\times 1=0.04$
I	$0.038+(0.04-0.038)\times 0.499 =0.0389$	$0.038+(0.04-0.038)\times 0.665 =0.039$
A	$0.038+(0.039-0.038)\times 0.665 =0.038$	$0.038+(0.039-0.038)\times 1=0.039$

BEJAIA : [0.038 0.039[

d) Codage LZW

Le codage LZW (Lempel–Ziv–Welch) est une technique de codage sans perte. Cette technique de compression est utilisée en GIF, TIFF et PDF. Le principe est de chercher des modèles qui se répètent dans une séquence de données.

L'algorithme lit une séquence de symboles puis regroupe en bloc puis attribut pour chaque bloc un code.

Le principe de la compression est le suivant :

1. On crée une table où la première colonne corresponde au mot (lettre lue) à coder, si le mot est déjà mentionné précédemment on lui ajoute une lettre sinon, c'est un nouveau mot donc on l'ajoute a la table des symboles codés.
2. La deuxième colonne correspond au code des symboles donnés (exemple : code ASCII)
3. La troisième colonne correspond à l'emplacement du code (si on utilise la table du code ASCII, on commence de 257 vu que les codes ASCII occupent les emplacements 0 à 256).

Exemple: soit la table de code suivant :

Symbole	code
□	1
A	2
B	3
O	4
W	5

Exemple 2 : On veut coder la séquence: **wabba□wabba□wabba□**

Symbole	Code	Nouveau code
W	5	
Wa		6
A	2	
Ab		7
B	3	
Bb		8
B	3	
Ba		9
A	2	
a□		10
□	1	
□ wa		11
Wa	6	
Wabb		12
Bb	8	
bba□		13
a□	10	
a□wabb		14
Wabb	12	
Wabba□		15
a□	10	

Décodage

Le principe du décodage est le suivant:

- On lit le premier code dans le dictionnaire et on le remplace par le mot qui lui correspond dans la table.
- Au fur et à mesure qu'on lit les codes on ajoute au premier mot décodé.

Le tableau suivant montre les différentes étapes de la décompression

Code	Mot lu	Mot décodé	Mot ajouté au dictionnaire
5	W	W	
2	A	Wa	Wa, 6
3	B	Wab	
3	B	Wabb	
2	A	wabba	
1	□	wabba□	
6	Wa	wabba□ wa	
8	Bb	wabba□ wabb	Bb,8
10	a□	wabba□ wabb a□	a□, 10
12	wabb	wabba□ wabb a□ wabb	Wabb, 12
10	a□	wabba□ wabb a□ wabb a□	

2.4 Efficacité d'un code

Pour chaque source sans mémoire il existe un code préfix d'efficacité

$$E = \frac{H(A)}{L_{moy}} \quad (2.8)$$

$H(A)$: entropie de la source A

L_{moy} : longueur moyenne du code

2.5 Conclusion

Le codage de source sert à compresser les données. Il existe différents types de codage de source, tels que le codage de Huffman, le codage arithmétique, le codage Lempel-Ziv. Le codage de source est utilisé pour de nombreuses applications, telles que la compression de fichiers audio, vidéo et image, la transmission de données sur des réseaux à bande passante limitée, et la sauvegarde de données.

Chapitre 3 : Codage Canal

3.1. Introduction

Les données transmises sur un canal, subissent des altérations et perturbations dues aux différentes sources de bruits. En effet, les bruits peuvent être causés par le canal lui-même ou de différents dispositifs électroniques qui constituent le système de transmission.

Pour une communication fiable, des chercheurs ont proposé plusieurs codages canal dans le but de détecter d'éventuels bruits et de les corriger

3.2. Schéma simplifié d'un système de communication

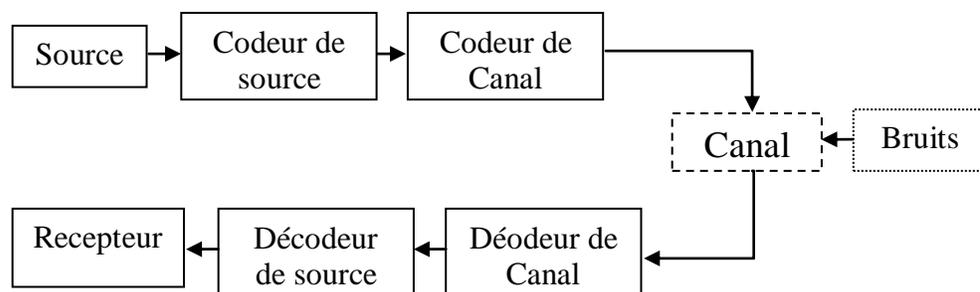


Figure 3.1 Un schéma général du système de la communication selon Shannon

Source d'information (données): Audio, image, vidéo, texte, . . .

Codeur de source: code et compresse les données à transmettre.

Canal: constitué d'un codeur pour corriger d'éventuels bruits et un support de transmission (fibre optique, support magnétique ou optique, ...)

Bruit : perturbations électromagnétiques, rayures, . . .

Avant la réception les données transmises sont décodées avec premièrement un décodeur canal suivi d'un décodeur de source.

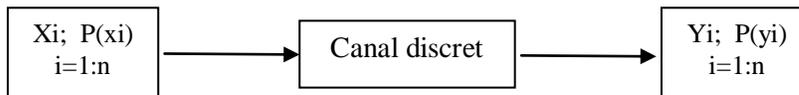
3.3. Modélisation d'un canal

Le théorème de Shannon stipule que pour qu'il ait une transmission fiable sur un canal, qui peut subir des bruits, il faut que le débit binaire du message transmis ne dépasse pas un certain débit appelé **capacité du canal**.

Supposant que la variable X représente l'entrée du canal et Y la sortie du canal. La loi de transition entre X et Y est une loi conditionnelle $P_{Y/X}$.

Par définition : $P: X * Y \rightarrow [0, 1]$ tel que :

$$\forall x \in X, \sum_{y \in Y} P(y/x) = 1 \quad (3.1)$$



Les sorties Y_i du canal sont reliées aux entrées X_i par une matrice de probabilité ou matrice de transition du canal A

$$\begin{bmatrix} p(y_1) \\ p(y_2) \\ p(y_3) \\ \vdots \\ p(y_n) \end{bmatrix} = A \begin{bmatrix} p(x_1) \\ p(x_2) \\ p(x_3) \\ \vdots \\ p(x_n) \end{bmatrix} \quad (3.2)$$

Cette égalité montre que les probabilités de sorties sont des combinaisons linéaires des probabilités d'entrées.

- *Le calcul de la matrice de transition du canal A*

Selon la règle de Bayes, la probabilité conjointe $P(x_i, y_j)$ c'est la probabilité d'émission de x_i fois la probabilité de réception de y_j sachant x_i a été émis.

$$P(x_i, y_j) = p(x_i) \cdot p(y_j/x_i) \quad (3.3)$$

$$\sum_{i=1}^n P(x_i, y_j) = \sum_{i=1}^n p(x_i) \cdot p(y_j/x_i) \quad (3.4)$$

Sachant que :

$$\sum_{i=1}^n P(x_i, y_j) = p(y_j) \quad (3.5)$$

$$p(y_j) = \sum_{i=1}^n p(x_i) \cdot p(y_j/x_i) \quad (3.6)$$

$$p(y_j) = [p(x_1) \ p(x_2) \ p(x_3) \ \dots \ p(x_n)] \begin{bmatrix} p(y_j/x_1) \\ p(y_j/x_2) \\ p(y_j/x_3) \\ \vdots \\ p(y_j/x_n) \end{bmatrix} \quad (3.7)$$

D'où :

$$A = \begin{bmatrix} [p(y_1/x_1) \ p(y_2/x_1) \ \dots \ p(y_m/x_1)] \\ [p(y_1/x_2) \ p(y_2/x_2) \ \dots \ p(y_m/x_2)] \\ [p(y_1/x_3) \ p(y_2/x_3) \ \dots \ p(y_m/x_3)] \\ \vdots \\ [p(y_1/x_n) \ p(y_2/x_n) \ \dots \ p(y_m/x_n)] \end{bmatrix} \quad (3.8)$$

3.4. Types de canaux et leurs capacités

3.4.1. Le canal binaire symétrique (BSC pour Binary Symmetric Channel)

C'est un canal binaire, c'est-à-dire il y a deux messages à transmettre. La probabilité du message $p(x_1) = p$ et le deuxième message $p(x_2) = 1 - p$. Et à la réception le message y_1 de probabilité $p(y_1) = q$ et le message y_2 de probabilité $p(y_2) = 1 - q$.

Construisant la matrice de transmission

$$\begin{bmatrix} q \\ 1 - q \end{bmatrix} = A \begin{bmatrix} p \\ 1 - p \end{bmatrix} \quad (3.9)$$

$$A = \begin{bmatrix} p(y_1/x_1) & p(y_2/x_1) \\ p(y_1/x_2) & p(y_2/x_2) \end{bmatrix} \quad (3.10)$$

$p(y_1/x_2)$: Probabilité de recevoir y_1 tel que x_2 est émis est une erreur de transmission.

$p(y_2/x_1)$: Probabilité de recevoir y_2 tel que x_1 est émis est une erreur de transmission.

La matrice de transmission A est dite **symétrique** si les erreurs de transmission sont égales

$$\text{Erreur} = e = p(y_1/x_2) = p(y_2/x_1) \quad (3.11)$$

La matrice A peut s'écrire en fonction de la probabilité d'erreurs tel que

$$A = \begin{bmatrix} 1 - e & e \\ e & 1 - e \end{bmatrix} \quad (3.12)$$

Le canal BSC se schématise:

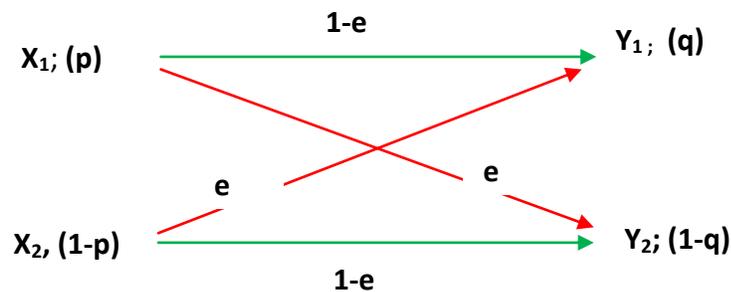
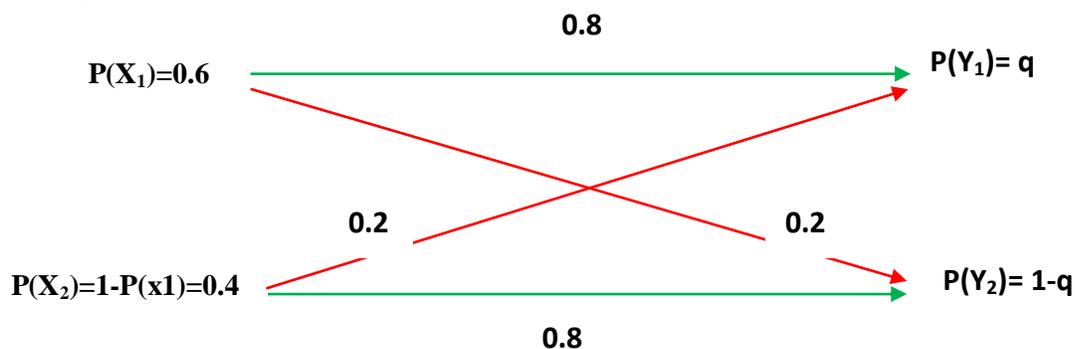


Figure 3.2 schéma du canal BSC.

Exemple :



$$\begin{bmatrix} P(y1) \\ P(y2) \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}^T \begin{bmatrix} P(x1) \\ P(x2) \end{bmatrix}$$

Où

$$[P(y1) \ P(y2)] = [P(x1) \ P(x2)] \begin{bmatrix} P(y1/x1) & P(y2/x1) \\ P(y1/x2) & P(y2/x2) \end{bmatrix}$$

$$[P(y1) \ P(y2)] = [0.6 \ 0.4] \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$$

$$[P(y1) \ P(y2)] = [0.34 \ 0.66]$$

a) La capacité d'un canal binaire symétrique (BSC)

La capacité C d'un canal binaire symétrique est le maximum la quantité d'information $I(x, y)$ lorsqu'on fait varier les probabilités des entrées (information à transmettre) $\{p(x_k)\}_k$

$$C = \max_{\{p(x_k)\}_k} (I(x, y)) \quad (3.13)$$

b) Calcul de C

$$I(x, y) = H(Y) - H(Y/X) \tag{3.14}$$

$$I(x, y) = H(X) - H(X/Y) \tag{3.15}$$

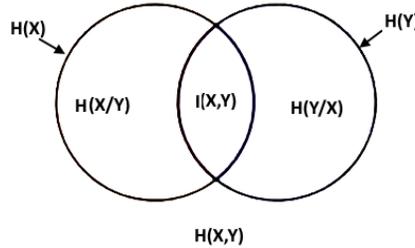


Figure 3.3 Information mutuelle entre X et Y

$I(X;Y)$: Information mutuelle entre X et Y

$H(X)$: Entropie de la source X

$H(X/Y)$: Entropie conditionnelle de X sachant Y

$H(Y/X)$: Entropie conditionnelle de Y sachant X

Selon les probabilités de sortie (**Figure 3.2**) :

$$1- H(Y) = q \log_2 \left(\frac{1}{q} \right) + (1 - q) \log_2 \left(\frac{1}{1-q} \right) \tag{3.17}$$

$$q = p(1-e)+e(1-p) \tag{3.18}$$

2- Selon la règle de Bayes appliquée aux entropies

$$H(Y/X) = p H(Y/X1) + (1 - p)H(Y/X2) \tag{3.19}$$

$$H(Y/X1) = H(Y/X2) = e. \log_2 \left(\frac{1}{e} \right) + (1 - e). \log_2 \left(\frac{1}{1-e} \right) \tag{3.20}$$

Finalemnt

$$I(x, y) = q. \log_2 \left(\frac{1}{q} \right) + (1 - q). \log_2 \left(\frac{1}{1-q} \right) - e. \log_2 \left(\frac{1}{e} \right) - (1 - e). \log_2 \left(\frac{1}{1-e} \right) \tag{3.21}$$

On dérive pour trouver le maximum

$$\frac{\partial I(X,Y)}{\partial p(x_k)} = \frac{dI(X,Y)}{dp} = \frac{dI(X,Y)}{dq} \cdot \frac{dq}{dp} \tag{3.22}$$

$$\frac{dI(X,Y)}{dq} = (1 - 2e) \cdot \log_2\left(\frac{1}{q} - 1\right) \quad (3.23)$$

$$\frac{dI(X,Y)}{dq} = 0 \quad \text{si } q=1/2, \text{ on replace dans (3.18)}$$

$$p(1 - e) + e(1 - p) = 1/2 \rightarrow q = 1/2 \quad (3.24)$$

$q = p$: veut dire que les messages transmis sont équiprobables et donc les messages reçus sont également équiprobables.

Finalement la capacité du canal est la quantité de l'information mutuel lorsque la probabilité à l'émission $p=1/2$

$$C = I(x, y)_{p=\frac{1}{2}} = 1 + e \cdot \log_2(e) + (1 - e) \log_2(1 - e) \quad (3.25)$$

Cas particulier

Si $e=1/2$ (les deux bits ont les mêmes chances d'être reçu) $\rightarrow q=1/2 \rightarrow I(x,y)=0$

Donc $C=0$, toute transmission comporte des erreurs.

La courbe de capacité en fonction de e

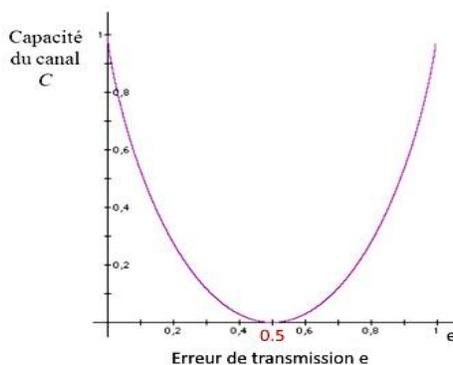


Figure 3.4 Erreur de transmission Canal Binaire Symétrique

On distingue trois cas :

- 1) $e = 0 \rightarrow C(0)=1\text{bit/s/Hz} \rightarrow$ transmission sûre
- 2) $e=1/2 \rightarrow C(1/2)=0\text{bit/s/Hz} \rightarrow$ transmission pas sûre
- 3) $e=1 \rightarrow C(1)=1\text{bit/s/Hz} \rightarrow$ transmission sûre et les bits sont inversés (envoi 1 et reçoit 0)

3.4.2. Le Canal Binaire à Effacement BEC

Le canal BEC (*Binary Erasure Channel*) transmet l'un des deux bits binaires 0 et 1 soit reçu sans erreur. Cependant, un effacement «e» est produit lorsque le récepteur ne reçoit pas le bit envoyé.

Le BEC efface un bit avec une probabilité, appelée la probabilité du canal à effacement. Ainsi, les probabilités de transition de canal pour le BEC sont les suivantes:

Cas 1 : la sortie reçoit le bit d'entrée avec une probabilité de $1-e$

$$P(y = 0 / x = 0) = 1 - e$$

$$P(y = 1 / x = 1) = 1 - e$$

Cas 2: la probabilité de recevoir un (1-bit envoyé) est nulle

$$P(y = 1 / x = 0) = 0$$

$$P(y = 0 / x = 1) = 0$$

Cas 3: la sortie produit un effacement lorsque la sortie ne reçoit pas de bit envoyé

$$P(y = e / x = 0) = e$$

$$P(y = e / x = 1) = e$$

Le canal BEC se schématise:

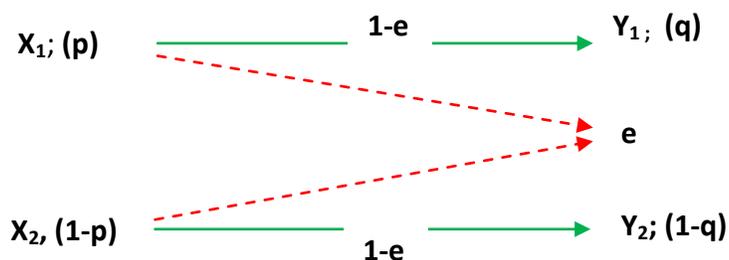


Figure 3.5 Schéma canal à effacement

D'où la matrice de transition :

$$\begin{bmatrix} 1 - e & e & 0 \\ 0 & e & 1 - e \end{bmatrix}$$

D'où

$$[y1 \quad y2 \quad y3]$$

$$\begin{bmatrix} x1 \\ x2 \end{bmatrix} \cdot \begin{bmatrix} 1-e & e & 0 \\ 0 & e & 1-e \end{bmatrix}$$

$$y1 = (1-e)x1 + 0 \cdot x2$$

$$y2 = e \cdot x1 + e \cdot x2$$

$$y3 = 0 \cdot x1 + (1-e) \cdot x2$$

a) La capacité d'un Canal Binaire à effacement BEC

$$C_p = \max_{\text{lois de } X} I(X;Y) = \max_{\text{lois de } X} [H(Y) - H(Y/X)]$$

$$H(Y/X) = P(X=0)[-p\text{Log}_2 p - (1-p)\text{Log}_2(1-p)] + P(X=1)[-p\text{Log}_2 p - (1-p)\text{Log}_2(1-p)]$$

Pour le cas où $P(X=0)=P(X=1)=0.5$, nous aurons : $H(Y/X) = -p\text{Log}_2 p - (1-p)\text{Log}_2(1-p)$

Nous allons calculer maintenant $H(Y)$:

$$P(Y=1) = P(X=1)(1-p) = q(1-p); P(Y=0) = P(X=0)(1-p) = (1-q)(1-p)$$

Où nous supposons que $P(X=1)=q$ et bien sûr $P(X=0)=1-q$

$$P(Y=\varepsilon) = (1-q)p + qp = p$$

$H(Y)$ est donc égale à :

$$H(Y) = -q \cdot (1-p) \cdot \text{Log}_2 q \cdot (1-p) - (1-q) \cdot (1-p) \cdot \text{Log}_2(1-q)(1-p) - p\text{Log}_2 p$$

$$I(X;Y) = H(Y) - H(Y/X) = -q(1-p)\text{Log}_2 q(1-p) - (1-q)(1-p)\text{Log}_2(1-q)(1-p) - p\text{Log}_2 p + p\text{Log}_2 p - (1-p)\text{Log}_2(1-p) = (1-p)[-q\text{Log}_2 q - (1-q)\text{Log}_2(1-q)]$$

b) Calcul de C

$$C_p = \max_{\text{lois de } X} I(X;Y)$$

Le maximum est obtenu pour une loi de probabilité sur X, uniforme, alors

$$-q\text{Log}_2 q - (1-q)\text{Log}_2(1-q) = 1$$

Et la capacité :

$$C_p = 1-p$$

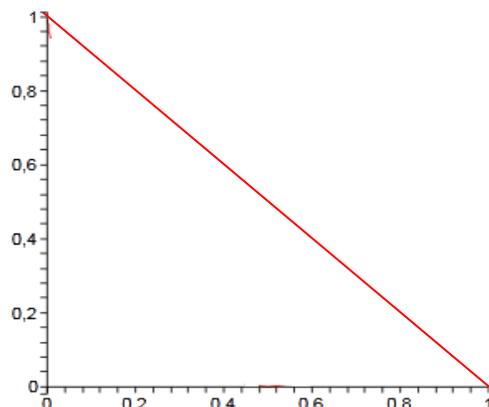


Figure 3.6 Capacité d'un canal à effacement en fonction de p.

3.5. Codage correcteur d'erreurs

On distingue différentes familles de codes correcteurs d'erreurs:

3.5.1. Les codes de répétition

Est l'un des codes correcteurs d'erreurs les plus simples.

Exemple : supposant qu'on veut envoyer 1010.

- Avant l'envoi, les bits sont répétés un certain nombre fois

Exemple 3 fois : 111 000 111 000

- Si le récepteur détecte : 011 100 011 000

On vérifie pour chaque groupe de 3 bits, un bit est considéré comme une erreur s'il est différent des deux autres.

011 100 011 000

3.5.2. Les codes linéaires (Hamming)

La distance de Hamming entre deux mots binaires est le nombre de bits qui diffèrent entre les deux mots. Si X et Y sont deux mots de codes, La distance de Hamming est noté $d(X,Y)$

Exemple : $d(1010, 1100) = 2$

$$d(101010, 111100) = 3$$

Si x est le message envoyé et y le message reçu

Si $d(X,Y) = 0 \rightarrow$ pas d'erreur

Si plusieurs messages sont envoyés l'algorithme des K proches voisins est utilisé pour détecter le message. Le principe est de calculer les distances de Hamming entre le code reçu et tous les mots de code envoyés et en garde le code ou la distance est minimale.

Exemple:

Soit les mots de codes envoyés suivants : {000 111, 001 100, 111 101}

Et le mot reçu est : $Y = \{111 100\}$

$$d(Y, 000 111) = 5$$

$$d(Y, 001 100) = 2$$

$$d(Y, 111 101) = 1$$

Y est proche de 111 101 donc c'est message envoyé

Nombre d'erreurs qu'on peut détecter et corriger

- La distance minimum d'un code C est la distance la plus petite entre chaque deux mots de code.

Exemple : soit e code suivant qui contient trois mots de codes $C : \{000 111, 001 100, 111 101\}$

Calculant toutes les distances possibles :

$$d(000 111, 001 100) = 4$$

$$d(000 111, 111 101) = 4$$

$$d(001 100, 111 101) = 3$$

La distance minimum est 3

- **Le nombre d'erreurs qu'on peut détecter**

Supposant que la distance minimum de Hamming d'un code est $2e + 1$

Alors :

Le nombre d'erreurs qu'on peut détecter est : **$2e$ erreurs**

Le nombre d'erreurs qu'on peut corriger est : **e erreurs**

Exemple 1: si la distance minimal est 15

Donc : $2e+1=15$

On peut détecter $2e=$ **14 erreurs**

on peut corriger : $e=(15-1)/2 =$ **7 erreurs**

Exemple 2: si la distance minimal est 4

Donc : $2e+1=4$

On peut détecter $2e=$ **3 erreurs**

on peut corriger : $e=(4-1)/2 =1,5 \rightarrow$ **1 erreurs**

3.5.3. Code linéaire

a) Code (n, k) -code

- Soit F un espace vectoriel, F_2 est un corps à deux éléments 0 et 1. Les mots de longueur n , composés de deux éléments (0 et 1) sont des codes de l'espace vectorielle F_2^n .

- Un code linéaire C de longueur n et de dimension 2^k est un sous-espace vectoriel ($C \subset F_2^n$).
Le nombre de mots du code C est 2^k , on dit que C est un **(n, k) -code**

Exemple :

$C=\{00000, 01010\}$

-Le cardinal de C (nombre de mots) $|C| =2 =2^1$ donc $K=1$

-Tous les codes sont de longueur 5 \rightarrow c'est un code de $n=5$ -codes

Donc c'est un **$(5,1)$ -code**

b) Code systématique

Un code systématique est un (n, k) – code tel que pour tous les mots binaires de longueur K sont préfixes de (n, k) – code.

Exemple:

soit $C = \{0010, 0111, 1010, 1101\}$

$|C| = 4 = 2^2$ donc $K=2$

Longueur de chaque mot est $n=4$ donc c'est un $(4,2)$ -code

Systematique ?

$K=2$, donc $2^2=4$ compositions possibles

00

01

10

11

On remarque que tous ces 4 mots sont préfixes des codes C , donc c'est un **$(4,2)$ -code systématique.**

c) Matrice génératrice d'un code systématique linéaire

- Soit un code $C = \{000000, 001110, 010101, 011011, 100011, 101101, 110100, 111010\}$

C'est un $(6, 3)$ -code systématique

$K=3$

La matrice identité $I_{K \times K}$

$$I_K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Soit

Soit les k premiers mots de codes $c = [c_1 c_2 \dots c_k]$, les $(k+1 \dots n)$ codes trouvés par combinaison linéaire des k premier code ainsi :

$$c_{k+1} = A_{11}c_1 \oplus A_{12}c_2 \oplus \dots \oplus A_{1k}c_k$$

$$\dots$$

$$c_n = A_{n1}c_1 \oplus A_{n2}c_2 \oplus \dots \oplus A_{nk}c_k$$

- La **matrice génératrice G** est la matrice utilisée pour encoder un code linéaire en ajoutant des bits de redondance, dont les lignes (mots) sont les codes dont les préfixes correspondent aux mots de la matrice identité $I_{3 \times 3}$

$$G = \begin{bmatrix} 1 & 0 & 0 & A_{11} & A_{21} & A_{31} \\ 0 & 1 & 0 & A_{12} & A_{22} & A_{32} \\ 0 & 0 & 1 & A_{1k} & A_{21} & A_{33} \end{bmatrix}$$

d) Encodage

Une matrice génératrice (K, n) permet de coder des codes multipliant un mot de longueur K pour avoir un code de dimension $(1, n)$. Où la multiplication est effectuée modulo 2.

Exemple : soit la matrice génératrice de l'exemple précédent. On souhaite trouver le code du mot : 101

$$C = [1 \ 0 \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$C = [1 \ 0 \ 1 \ 1 \ 0 \ 1]$$

- Coder le mot $m = [1 \ 1 \ 1]$

$$C = m \cdot G = [1 \ 1 \ 1 \ 0 \ 0 \ 0]$$

e) Matrice de contrôle H

La matrice de contrôle H permet de vérifier s'il y a des erreurs. Elle est construite à partir de la matrice G .

$$G_{K \times P} = [I_{K,K} \ A_{K,K-P}]$$

$$H = \begin{bmatrix} A_{K,(K-P)} \\ I_{K,K} \end{bmatrix}$$

Exemple:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \rightarrow H = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Si on envoie un message m et en reçoit le code $r=100111$

On calcule ce qu'on appelle le syndrome de r qui est $S= r.H$

$$r.H=[100111]. \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S = [1 \ 0 \ 0]$$

S n'est pas nul donc y a une erreur

La position de l'erreur correspond au chiffre S est converti en base décimale,

$(100)_2=(4)_{10}$ donc l'erreur est au 4eme bit de $r = \mathbf{100111}$

Donc le message transmit le code corrigé qui est : $r = \mathbf{100011}$

- Il convient d'appeler vecteur message \vec{m} la matrice ligne dont les k éléments sont les bits ordonnés du message

$$\vec{m} = (m_1, m_2, \dots, m_k) \quad (3.15)$$

On appelle de même vecteur de parité \vec{P} la matrice ligne dont les r éléments sont les bits de parité ordonnés

$$\vec{P} = (p_1, p_2, \dots, p_r) \quad (3.16)$$

Et on note \vec{c} le vecteur constitué des n bits constituant le mot codé

$$\vec{c} = (c_1, c_2, \dots, c_n) \quad (3.17)$$

3.6. Conclusion

Le codage canal est fondamental dans la chaîne de transmission numérique qui permet d'assurer une transmission fiable et efficace des données sur des canaux de communication bruyants. En introduisant une redondance dans les données transmises, le codage détecte et corrige les erreurs qui se produisent pendant la transmission. Il existe différents types de schémas de codage canal.

Chapitre 4 Méthodes de compression avec pertes

4.1 Introduction

Contrairement à la compression sans perte (Empirique, Huffman, ...), la compression **avec perte ne permet pas de restituer** le signal où les données envoyées exactement comme elles étaient à l'origine. Les données décompressées se rapprochent des données originales selon la qualité désirée

Ce type de compression est utilisé pour les images, le son et la vidéo. Les formats de données tels que JPEG, MP3, MP4,...

La compression avec pertes est appliquée généralement pour les données perceptuelles, comme des fichiers audio, image et vidéo. La réduction de l'information dans les données n'est pas souvent perceptible par le système visuel ou auditif de l'être humain, la compression avec perte est basée sur ces deux dernières caractéristiques humaines.

4.2 Compression basées sur les transformations

Les algorithmes de compression avec perte comportent plusieurs étapes telles que la transformation, la quantification et le codage entropique. Le schéma de la figure 4.1 les étapes schéma général des méthodes de compression avec perte basées sur la transformation.

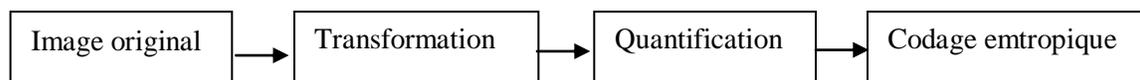


Figure 4.1 les étapes de compression avec perte d'une image

4.2.1 Transformation

Le but de la transformation est d'une part de dé-corréler des coefficients et d'autre part de regrouper les coefficients qui ont les mêmes caractéristiques (fréquence, entropie,...). Les transformations peuvent être :

- Transformée en ondelette (DWT)
- Transformée en cosinus (DCT)
- Transformée en Fourier Discrète (DFT)
- ...

4.2.2 Quantification

La quantification est une étape très importante dans la chaîne de compression. Elle permet de supprimer ou de garder certains coefficients de la transformée, c'est l'étape la perte de données est effectuée.

Exemple de quantification d'un signal 1D

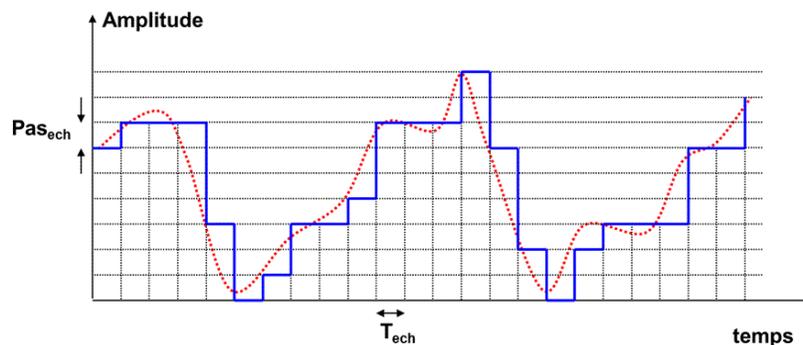


Figure 4.2 Quantification d'un signal 1D

4.2.3 Codages entropiques

Codage entropique

Codage de Huffman

Codage de Shannon-Fano

Codage arithmétique

4.3 Critère d'évaluation

Le critère d'évaluation permet de mesurer la détérioration ou la dégradation générée par la compression. Il existe plusieurs critères tels que MSE, PSNR, CR, SSIM ..etc

4.3.1 MSE(Mean Square Error)

Mesure l'erreur quadratique moyenne (EQM ou MSE). Soit une image originale I de taille $(M \times N)$ et \hat{I} est sa version compressée.

Le MSE est calculé :

$$EQM = \frac{1}{M \cdot N} \sum_i^N \sum_j^M (I_{i,j} - \hat{I}_{i,j})^2 \quad (4.1)$$

4.3.2 Peak Signal to Noise Ratio (PSNR)

C'est un critère qui mesure le rapport entre la puissance du signal utile par la puissance du bruit.

La mesure du rapport signal sur bruit (SNR) en décibel (db) est calculée par l'expression suivante :

$$SNR = 10 \log_{10} \left(\frac{\sum_{i=1}^M \sum_{j=1}^N I^2(i, j)}{\sum_{i=1}^M \sum_{j=1}^N [I(i, j) - \hat{I}(i, j)]^2} \right) \quad (4.2)$$

I : Image original de taille $(M \times N)$

\hat{I} : Image compressée.

$$PSNR_{db} = 10 \cdot \log_{10} \left(\frac{L^2}{EQM} \right) \quad (4.3)$$

L : nombre d'échelons de Luminance (255 pour 8 bits)

4.3.3 Structural similarity index measure (SSIM)

Est une **mesure de similarité entre deux images numériques**. Le principe est d'évaluer la dégradation en fonction du contexte local du défaut ; il est fait une pondération de 3 paramètres : Luminance, Contraste et Contours :

$$SSIM(x, y) = \frac{(2\mu_x\mu_{\hat{x}}+C_1)(2Cov_{x\hat{x}}+C_2)}{(\mu_x^2+\mu_{\hat{x}}^2+C_1)(\sigma_x^2+\sigma_{\hat{x}}^2+C_2)} \quad (4.4)$$

μ_x : Moyenne de x

$\mu_{\hat{x}}$: Moyenne de \hat{x}

σ_x : Ecart type de x

$\sigma_{\hat{x}}$: Ecart Type de \hat{x}

$Cov_{x\hat{x}}$: Covariance de $x\hat{x}$

$C_1 : (k_1L)^2$, avec $k_1 = 0,01$ et L: nombre d'échelons de luminance

$C_2 : (k_2L)^2$, avec $k_2 = 0,03$ et L: nombre d'échelons de luminance

La mesure SSIM donne une valeur comprise entre -1 et 1, où une valeur de 1 indique que les deux images sont parfaitement similaires et une valeur de -1 indique que les deux images sont complètement différentes. Une valeur de 0 indique une absence de corrélation entre les deux images.

La mesure SSIM est une méthode efficace pour évaluer la qualité d'image en évaluant la similarité structurelle entre deux images, et est largement utilisée dans le domaine du traitement d'images pour évaluer la qualité de l'image reconstruite ou compressée par rapport à l'image d'origine.

Conclusion

Les techniques de compression avec perte sont utilisées dans les applications de médias numériques, elles réduisent les données au niveau acceptable de qualité perceptuelle.

Il existe plusieurs méthodes de compression avec perte, notamment le codage par transformation, telles que la transformée en cosinus discrète (DCT) ou la transformée en ondelette discrète (DWT), pour convertir le signal d'origine en un ensemble de coefficients qui peuvent être quantifiés et codés plus efficacement.

Chapitre 5 Techniques de compression d'images (Cas du JPEG)

5.1 Introduction

JPEG veut dire Joint Photographic Experts Group, c'est un groupe d'experts qui développe et standardise les normes de compression images. Le groupe JPEG à l'aide des experts internationaux, a établi la norme en 1991 et adopté officiellement en 1992.

JPEG est aussi le nom de l'algorithme de compression. Plusieurs extensions utilisent la compression JPEG tel que jpg et .jpeg, jpe.

5.2 Algorithme JPEG

L'algorithme de compression JPEG comporte 6 étapes : (1) Transformation de l'espace couleurs (2) le sous-échantillonnage (3) découper l'image en blocs (4) transformation DCT appliquée pour chaque bloc (5) quantification (6) codage RLE et Huffman.

La décompression comporte aussi six étapes : (1) Décodage RLE et Huffman (2) quantification inverse (3) DCT inverse (4) assemblage de blocs (5) sur-échantillonnage (6) transformation de couleur.

La figure 5.1 montre les différentes étapes de compression et de décompression JPEG.

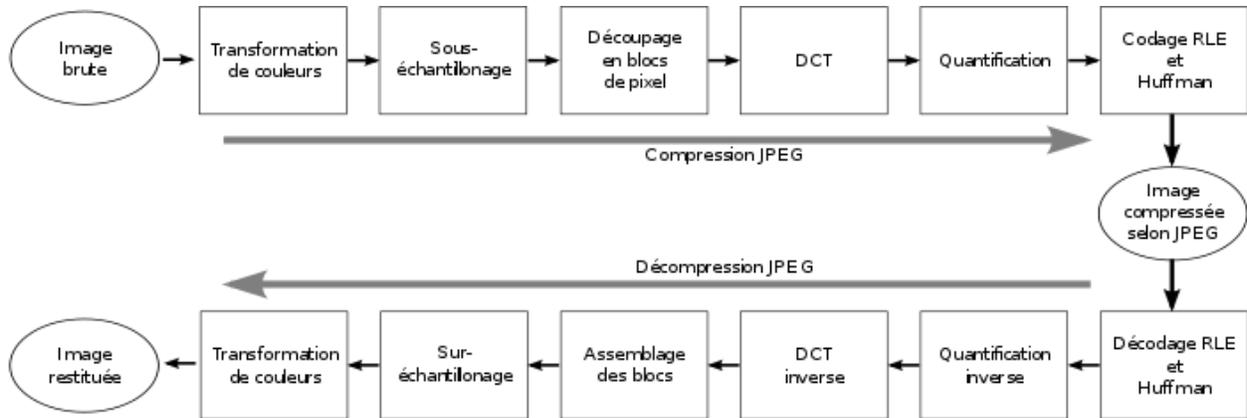


Figure 5.1 Algorithme de compression décompression de l'image norme JPEG

5.2.1 Transformation de couleur

L'image RVB (Rouge, Vert et Bleu) est transformée à l'espace couleur YCbCr (Y : luminosité, Cb : chrominance bleue et Cr pour la chrominance rouge) et en soustrait 128 des composantes (Y, Cb et Cr) afin d'avoir la gamme de valeur entre -128 et 127. Ensuite les trois composantes sont traitées séparément.

$$Y = 0.299R + 0.587G + 0.114B \quad (5.1)$$

$$Cr = 0.701R - 0.587G - 0.114B \quad (5.2)$$

$$Cb = -0.299R - 0.587G + 0.866B \quad (5.3)$$



(a)

(b)

Figure 5.2 Transformation d'image RGB vers YCbCr

L'espace couleur YCbCr réduit la taille de l'image originale, et c'est un espace qui sépare l'information couleur de la luminance (Y). Et comme l'œil humain est sensible à la luminance qu'à la chrominance on peut sous échantillonner les plans chrominances Cb, Cr, d'où l'intérêt de l'espace YCbCr.

5.2.2 Sous échantillonnage

On va donc représenter une image en utilisant 3 tableaux, un par composante, par exemple:

Y				Cb				Cr			
141	150	143	147	170	172	174	179	85	84	79	83
142	150	147	144	169	170	168	179	84	80	82	86
140	139	143	145	169	171	173	175	82	89	85	84
138	145	150	148	171	172	170	170	82	87	86	83

Comme expliqué précédemment, l'œil humain n'étant pas sensible à la chrominance, les deux plans Cb et Cr sont sous échantillonnés. On calcule la moyenne de chaque bloc 2x2 on obtient les nouveaux tableaux (le Y n'est pas modifié) :

Y				Cb				Cr			
141	150	143	147	170	170	175	175	83	83	83	83
142	150	147	144	170	170	175	175	83	83	83	83
140	139	143	145	171	171	172	172	85	85	85	85
138	145	150	148	171	171	172	172	85	85	85	85

On ne garde qu'une seule valeur sur les 4 identiques :

Y			
141	150	143	147
142	150	147	144
140	139	143	145
138	145	150	148

Cb	
170	175
171	172

Cr	
83	83
85	85

Cet exemple montre que la chrominance est divisée par 4 avec le sous échantillonnage

5.2.3 Transformation en Cosinus Discret (DCT)

Chaque plan est divisé en blocs de 8x8 pixels, puis on applique la transformée en cosinus discrète DCT (Discrete Cosine Transform).

$$DCT(u, v) = \frac{2}{N \times M} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I(x, y) \cos \left(\frac{(2x+1)u\pi}{2N} \right) \cos \left(\frac{(2y+1)v\pi}{2M} \right) \quad (5.4)$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } u = 0, v = 0 \\ 1 & \text{si } u, v > 0 \end{cases} \quad (5.5)$$

La DCT correspond à une projection sur la base : bases fréquence et hautes fréquences, en d'autres termes la DCT permet de séparer les basses fréquences et les hautes fréquences de l'image.



Figure 5.3 Transformée en cosinus de l'image (a) image originale (b) sa DCT

5.2.4 Quantification

La matrice obtenue par DCT est quantifiée, c'est l'étape clé qui réduit l'information, donc perte de la qualité visuelle mais permet de gagner en espace disque (compression d'image).

- La quantification consiste à diviser cette matrice 8x8 (DCT) par une autre, appelée matrice de quantification, et qui contient 8x8 coefficients spécifiquement choisis par le codeur. Le but est ici d'atténuer les hautes fréquences, c'est-à-dire celles auxquelles l'œil humain est très peu sensible.

$$TQ(i, j) = 1 + (i + j + 1)K \quad (5.6)$$

Avec : **i** l'indice de ligne, **j** l'indice de colonne et **K** le facteur de qualité (choisi entre 1 et 25).

$$DCTQ(i, j) = \frac{DCT(i, j) \pm \frac{TQ(i, j)}{2}}{TQ(i, j)} \quad (5.7)$$

Le signe de $\pm \frac{TQ(i, j)}{2}$ suit le signe de la valeur $DCT(i, j)$

5.2.5 Balayage en séquence zigzag

La séquence zigzag est une technique qui permet de transformer une image (donnée matricielle) en un format linéaire dans le but de minimiser la quantité de bits nécessaires pour stocker et transmettre ces données.

La technique consiste à parcourir la matrice de l'image en zigzag, en commençant par la valeur en haut à gauche, en progressant diagonalement vers le bas à gauche jusqu'à la fin de la première ligne, puis en remontant diagonalement vers la droite jusqu'à la fin de la deuxième ligne, et ainsi de suite jusqu'à ce que toutes les valeurs de la matrice soient parcourues. Cette séquence en zigzag permet de regrouper les valeurs de pixels similaires qui se suivent dans l'image, ce qui facilite la compression de ces données.

La figure 5.4 montre le parcours de la séquence zigzag sur une image.

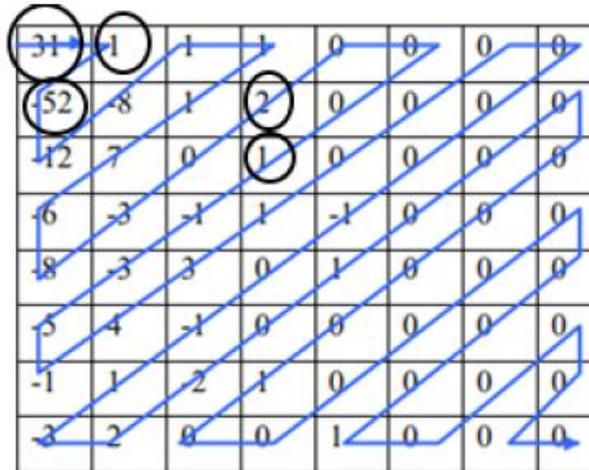


Figure 5.4 Scan en zigzag de l'image

5.2.6. Compression RLE

Le principe de la compression RLE est de représenter les données répétitives par une seule valeur suivie du nombre de fois où cette valeur se répète. Selon le parcours de la séquence zigzag, la compression RLE consiste à remplacer les pixels consécutifs nuls (zéro) par une valeur, qui est le nombre de fois qu'il se répète sans interruption.

[Nombre de zéros qui se suivent, valeur non nulle]

Les exemples entourés sur la matrice de la figure 5.4 : [0,31], [0,1], [0, -52].....[1, 2] ...[3,1]

Puis on code avec Huffman les statistiques trouvées par l'algorithme RLE (Nombre de zéros qui se suivent).

5.3 Compression d'images à base d'ondelette et Algorithme EZW

5.3.1 Transformée en ondelette

La transformée en ondelettes permet une analyse d'image à multi-résolutions/multifréquences. Cette décomposition de l'image en multifréquences permet d'avoir les changements brusques et les détails dans l'image (hautes fréquences) et les zones où il y a moins

de changement (basses fréquences) séparément. La figure 5.5 montre la décomposition de l'image à multirésolution.

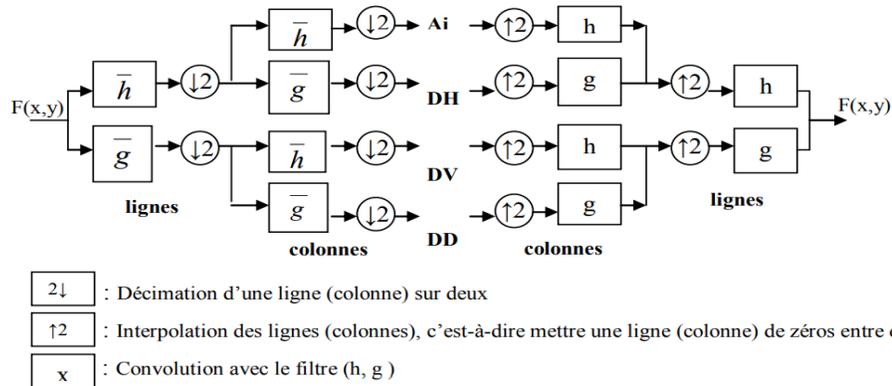


Figure 5.5 Décomposition de l'image à multi-résolutions

La figure 5.6 montre un exemple de la décomposition de l'image en sous-bandes qui contiennent les coefficients qui représentent les fréquences spatiales horizontales et verticales de l'image.



Figure 5.6 Décomposition d'une image en ondelette

La compression d'image par transformation en ondelettes est utilisée à plusieurs niveaux de décompositions et différentes ondelettes. Elle inclut les étapes suivantes:

1. La transformation de l'image en ondelettes.
2. La quantification.

3. La sélectionner les coefficients significatifs et les non-significatifs (supérieurs/inférieurs à un certain seuil).
4. Et enfin le codage des coefficients restants.

5.3.2 Compression d'image avec l'Algorithme EZW

EZW (Embedded Zerotree Wavelet) est un algorithme de compression d'image proposé en 1993 par Jérôme Shapiro. L'algorithme est basé sur la transformée d'ondelette (Discret Wavelet Transform: DWT), et dit imbriqué (Embedded).

L'algorithme introduit prend en considération les redondances entre les différentes résolutions. Plus on passe à la haute fréquence plus les coefficients diminuent, donc offrent plus de détails insignifiants, d'où codage progressif de l'algorithme.

L'algorithme introduit l'arbre de zéro (zerotree) en reliant les coefficients de chaque sous-bande qui se suit et formant ainsi une arborescence.

Algorithme EZW

- 1- Transformation de l'image en ondelette
- 2- Suppression de coefficients non significatifs

Les valeurs absolues des coefficients inférieures à un certain seuil T_0 sont considérés non significatives et sont supprimées.

$$C = \begin{cases} \text{Significatif} & \text{si } |C| > T_0 \\ \text{non significatif} & \text{si } |C| < T_0 \end{cases} \quad (5.8)$$

Le seuil est calculé ainsi :

$$T_0 = 2^{\lfloor \log_2(\max(DWTimg(i,j))) \rfloor} \quad (5.9)$$

Tel que $\max(DWTimg(i, j))$: désigne le plus grand coefficient d'ondelette existant dans l'image à la position i et j .

Pour balayer tous les coefficients deux méthodes sont utilisées: 'Raster scan' et 'Morton scan' où la différence réside dans le parcours de sélection des coefficients pour chaque bande DWT comme le montre la figure 5.7. Le principe est de parcourir les coefficients en partant des coefficients élevés (fréquences faibles) vers les coefficients faibles (fréquences élevées).

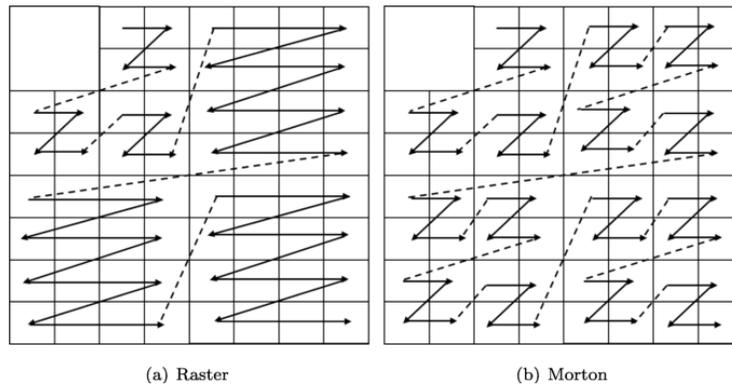


Figure 5.7 Deux différents scans de l'image (a) Scan Rater (b) Scan Morton

3- Le codage des coefficients

Après le scan et le classement des coefficients (significatif/ non significatif), l'algorithme classe les coefficients en quatre groupes selon signification et leurs signes :

- le coefficient 'C'est significatif avec un signe positif : Si $C > T_0$.
- le coefficient 'C'est significatif avec un signe négatif : Si $C < T_0$.
- le coefficient 'C'est non significatif, mais il existe un coefficient significatif parmi ses fils.
- le coefficient 'C'est non significatif, et il n'existe aucun coefficient significatif parmi ses fils.

Le tableau suivant les symboles du codage EZW ainsi que leurs indications

Symbole	Indication
Positif (P)	$ C > T_0$ et $C > 0$
Négatif (N)	$ C > T_0$ et $C < 0$
Zéro isolé (Z)	$ C < T_0$ et il existe un coefficient significatif parmi ses descendants (fils).
Zerotree (R)	$ C < T_0$ et il existe aucun coefficient significatif parmi ses descendants (fils).

4- La quantification par approximation successive

Cette étape fait réduire le seuil T_0 de moitié et refait la sélection des coefficients (significatifs/non significatifs).

$$T_1 = \frac{T_0}{2} \quad (5.10)$$

Les nouveaux coefficients sont ajoutés aux précédents.

Cette opération de sélection de coefficients peut être refaite avec un nouveau seuil $T_i = \frac{T_{i-1}}{2}$ tant que $T_i > 1$. Y a une transmission pour chaque seuil calculé, de cette façon on peut garder la qualité que l'on souhaite.

5- Codage des coefficients

Cette étape permet d'attribuer des 0 et 1 aux coefficients sélectionnés dans l'étape précédente.

On définit un autre seuil T_s tel que :

$$T_s = 3 * 2^{n-1} \quad (5.11)$$

Où n est choisi tel que : $T_0 = 2^n$

Les coefficients significatifs sont codés :

$$C = \begin{cases} 1 & \text{si } C > T_s \\ 0 & \text{si } C < T_s \end{cases} \quad (5.12)$$

5.4 Comparaison de JPEG et EZW

Le codage des coefficients comparaison des algorithmes les principaux avantages par rapport à JPEG sont :

- l'algorithme EZW s'adapte mieux aux propriétés locales de l'image.
- avec EZW on peut atteindre des taux de compression d'environ 50 contre 15 pour JPEG tout en ayant une qualité d'image raisonnable subjective.

5.5 Conclusion

Le JPEG (Joint Photographic Experts Group) est l'une des techniques de compression d'image les plus populaires. Il utilise des techniques de compression avec perte pour réduire la taille des images tout en conservant une qualité d'image visuellement acceptable.

Le JPEG reste un choix populaire pour la compression d'images en raison de son efficacité et de sa compatibilité avec de nombreux systèmes. Les méthodes de compression d'image EZW suivent les mêmes étapes qu'une compression d'images avec perte avec différentes techniques.

CONCLUSION

Le codage et la compression sont deux techniques essentielles pour la transmission et le stockage de données numériques.

Le codage consiste à représenter des données sous forme de codes numériques, afin de faciliter leur traitement et leur transmission. Il existe le codage de source et le codage de canal. Le codage de source permet de réduire les données à transmettre et sécurise l'information. Il existe différents types de codage de source, tels que le codage de Huffman, le codage arithmétique, le codage Lempel-Ziv. Le codage de canal quant à lui permet de coder de façon à protéger les données des bruits de canal en ajoutant de la redondance qui sera utilisée lors du décodage pour corriger d'éventuels erreurs.

Selon le théorème de Shannon, les deux fonctions contradictoires du codage source et canal n'affectent pas les performances et l'efficacité du système global.

Il existe deux types de compression. Compression sans perte tel que Huffman, le codage arithmétique, le codage Lempel-Ziv et la compression avec pertes. Contrairement à la compression sans perte, la compression avec perte ne permet pas de restituer les données originales mais se rapprochent des données originales selon la qualité désirée. Il existe plusieurs méthodes de compression avec perte, notamment le codage par transformation, telles que la transformée en cosinus discrète (DCT) ou la transformée en ondelettes discrète (DWT).

Le JPEG (Joint Photographic Experts Group) est un format de compression avec perte. L'algorithme de compression JPEG implique plusieurs étapes, notamment la conversion de l'image en espace de couleur YCbCr, la séparation de l'image en blocs de 8x8 pixels, la transformation de ces blocs en domaines de fréquence, la quantification de ces domaines et la compression entropique.

REFERENCES

1. Claude E. Shannon. A mathematical theory of communication. Bell Syst. Tech. J., 27(3) :379–423, 1948.
2. M. Cover and J. A. Thomas, “Elements of information theory“, 2nd edition, Wiley Series in telecommunications and signal Processing, 2006.
3. M. Barlaud, C. Labit, “Compression et codage des images et des vidéos“, traité Collection IC2, Ed. Hermès, 319p, 2002.
4. K. Sayood, "Introduction to Data Compression, Third Edition", Elsevier Inc. 2006.
5. Olivier Rioul, "Théorie de l'information et du codage", Edit. Lavoisier, 2007.
6. N. Moreau, "Outils pour la compression des signaux: applications aux signaux audio", Collection
7. Télécom, Edition Lavoisier, Octobre 2009.
8. J. C., Moreira, P. G., Farrell, "Essentials of Error-Control Coding", JohnWiley and Sons, Ltd, 2006.
9. C. Berrou, "Codes et turbocodes", Springer-verlag France, 2007.
10. Pennebaker, William B., and Joan L. Mitchell. JPEG: Still image data compression standard. Springer Science & Business Media, 1992.
11. D.A. Huffman, "A method for the construction of minimum-redundancy codes", Proceedings of the I.R.E., septembre 1952, pp 1098-1102
12. Welch, T. A. (June 1984). « A technique for high-performance data compression», Computer, vol. 17, p. 8-19.
13. Thomas M. Cover, Joy A. Thomas, Elements of Information Theory, Wiley-Interscience, 2006