1. Introduction à la théorie des codes

La théorie des communications s'intéresse aux moyens de transmettre une information depuis la source jusqu'à un utilisateur à travers un canal. La source peut être de n'importe quel type: textes, images, sons, vidéos par exemple. Le canal peut être une ligne téléphonique, une liaison radio, un support magnétique ou optique. La théorie des codes traite de la forme de l'information elle-même quand elle doit être transmise, ou stockée.

Le codage représente l'ensemble des opérations effectuées sur la sortie de la source avant la transmission. Ces opérations peuvent être par exemple la modulation, la compression, le brouillage, l'ajout de redondance pour combattre les effets du bruit, ou encore l'adaptation à des contraintes de spectre. Elles ont pour but de rendre la sortie de la source compatible avec le canal. Enfin le décodeur doit être capable de restituer de façon acceptable l'information fournie par la source.

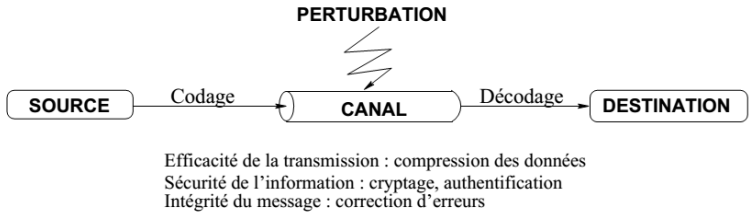


Fig. 1: Schéma fondamental du codage.

La théorie de l'information a été créée par C. E. Shannon dans les années 40. Il s'agit d'une théorie mathématique qui décrit les aspects les plus fondamentaux des systèmes de communication. Elle consiste en l'élaboration et l'étude de modèles pour la source et le canal qui utilisent différents outils. Pour des raisons de simplification, les modèles de sources et les modèles de canaux ainsi que leurs codages respectifs sont étudiés séparément.

- L'objectif du codeur de source est de représenter le message avec le moins de bits possibles. Pour ce faire, il cherche à éliminer toute la redondance contenue dans le message de la source.
- Le codage de canal a pour objet de protéger les données contre les erreurs qui peuvent survenir sur le canal physique lors de la transmission d'informations.

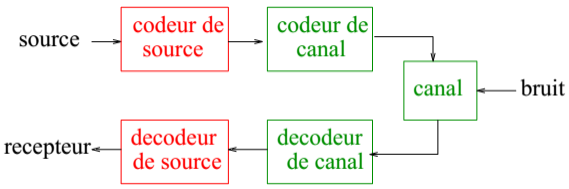


Fig. 2: Codeur de source et codeur de canal.

2. Sources et codage sources

La source génère un message à transmettre au destinataire. Celui-ci peut être analogique ou numérique. Dans un système de communication numérique, le message analogique devra être converti en numérique avant traitement.

2.1 Sources discrètes sans mémoire

La sortie d'une telle source est une séquence de lettres tirées dans un alphabet fini A = $\{a_1,...,a_n\}$. Chaque lettre de la séquence est choisie aléatoirement d'après une loi de probabilité p indépendante du temps. Pour toute lettre a, p(a) est la probabilité pour que cette lettre soit choisie. La donnée de $p(a_1),...,p(a_n)$ définit la probabilité discrète p sur A.

Exemple: Soit une source d'information qui fournit comme information l'une des quatre lettres a_1,a_2,a_3,a_4. Supposons que le codage de source transforme cette information discrète en symboles binaires. Nous donnons deux exemples de codage différents.

Codage 1	Codage 2
$a_1 \to 00$	$a_1 \to 0$
$a_2 \to 01$	$a_2 \to 10$
$a_3 \to 10$	$a_3 \to 110$
$a_4 \to 11$	$a_4 \to 111$

Si les quatre lettres sont équiprobables, la première méthode de codage est meilleure. Elle nécessite en effet deux symboles par lettre en moyenne tandis que la deuxième méthode nécessite 0.25 +0.25×2 +0.25×3+0.25×3 = 2, 25 symboles par lettres. En revanche, si l'on a une source dont la distribution de probabilités est $p(a_1) = 0.5$, $p(a_2) = 0.25$, $p(a_3) = p(a_4) = 0.125$, la longueur moyenne d'un symbole codé par la première méthode est toujours 2 tandis que celle d'un symbole codé par la deuxième méthode est 0.5 + 0.25×2 + 0.125×3 + 3 ×0.125 = 1.75. Le deuxième codage réussit donc à coder quatre symboles avec moins de deux bits. Il a réalisé une compression. Pour coder correctement une source, il est donc important de connaître son comportement statistique.

2.2 Sources analogiques

La sortie d'une telle source sera une fonction continue du temps, par exemple une tension qu'il faut coder par une séquence discrète binaire. La fonction continue doit être décrite le plus fidèlement possible par la séquence binaire générée par le codeur de source.

3. Canaux et codage de canal

Pour modéliser un canal de transmission, il est nécessaire de spécifier l'ensemble des entrées et l'ensemble des sorties possibles. La cas le plus simple est celui du canal discret sans mémoire. L'entrée est une lettre prise dans un alphabet fini A = $\{a_1,...,a_n\}$ et la sortie est une

lettre prise dans un alphabet fini B = {b_1,...,b_m}. Ces lettres sont émises en séquence et le canal est sans mémoire si chaque lettre de la séquence reçue ne dépend statistiquement que de la lettre émise de même position.

Ainsi un canal discret sans mémoire est entièrement décrit par la donnée des probabilités conditionnelles p(b|a) pour toutes les lettres a de l'alphabet d'entrée et toutes les lettres b de l'alphabet de sortie.

Exemple: Le plus connu est le canal binaire symétrique défini par A = B = {0, 1} et dont les probabilités de transition sont représentées par la figure ci-dessous. La probabilité pour qu'un symbole soit inchangé est 1−p et la probabilité pour qu'il soit changé est p.

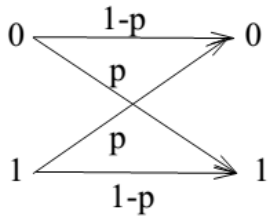


Fig. 3: Le canal binaire symétrique.

Il existe une classe de modèles de canaux appelés canaux continus, beaucoup plus proches des canaux physiques. L'entrée et la sortie sont alors des fonctions continues du temps.

4. Techniques de codage conjoint source/canal

Ce sont des techniques qui laissent volontairement de la redondance au niveau du codage de source ou qui effectuent l'opération de codage de source de manière unifiée avec l'insertion d'une protection sous forme de redondance pour contrer les aléas du canal.

5. Compression des données

Si le codage de canal ajoute de la redondance pour transmettre un signal en toute sécurité sur un canal bruité, la compression va tenter de retirer le plus de redondance possible d'une donnée. Les principales applications de la compression concernent l'archivage des données sur un disque dur, un CD ou un DVD. Certains formats intègrent directement de la compression (images JPEG ou fichier texte PDF). Dans le domaine de télécommunications, la compression des données est couramment utilisée dans le fonctionnement des modems (protocole V42 par exemple).

1. Rappels sur la théorie de l'information

Une information désigne par définition un ou plusieurs évènements possibles parmi un ensemble fini d'évènements. L'information permet de diminuer l'incertitude. Considérons par exemple une source qui peut produire trois symboles A, B et C. Quand le destinataire attend un symbole, il est dans l'incertitude quant au symbole que la source va engendrer. Lorsque le symbole apparait et qu'il arrive au destinataire, cette incertitude diminue. Le but de la théorie de l'information est de mesurer cette incertitude avant réception.

1.1 Mesure de l'information

Nous allons donner une mesure de la quantité d'information qui est adaptée à la description statistique des sources et des canaux. Les énoncés qui en résultent font appel aux probabilités discrètes. Nous allons en rappeler les notions principales.

1.1.1 Espace probabilisé discret

Nous considérerons des ensembles finis munis d'une probabilité discrète p. L'espace probabilisé est noté (A, p). La loi de probabilité est dite uniforme si p(a) = 1/n , où n = card(A), pour toute lettre a de A. Une variable aléatoire de (A, p) est une fonction de A dans un ensemble quelconque.

1.1.2 Probabilités conjointes et probabilités conditionnelles

Pour modéliser un canal discret, nous considérons l'espace A×B produit des deux ensembles $A = \{a_1, \ldots, a_n\}$ et $B = \{b_1, \ldots, b_m\}$. Le produit est formé des couples (a, b) avec a dans A et b dans B. On munit cet ensemble d'une loi de probabilité discrète, notée p_{AB}, appelée loi de probabilité jointe de A et B. La probabilité $p_{AB}(a, b)$ est la probabilité d'avoir simultanément a en entrée et b en sortie. On définit une loi de probabilité p_A sur A par:

$$p_A(a) = \sum_{b \in B} p_{AB}(a,b) \quad (1)$$

On définit une probabilité p_B sur B de façon similaire. Les deux lois p_A et p_B sont appelées lois marginales.

Nous définissons maintenant les lois conditionnelles. Soit a une lettre de A telle que p(a) > 0. La probabilité conditionnelle pour que l'on ait b en sortie sachant que l'on a en entrée est définie par:

$$p_{B|A}(b \mid a) = \frac{p_{AB}(a,b)}{p_A(a)} \qquad (2)$$

On dit également qu'il s'agit de la probabilité conditionnelle pour que l'on ait {B=b} sachant que {A=a}. De façon symétrique, on a:

$$p_{A|B}(a \mid b) = \frac{p_{AB}(a,b)}{p_B(b)} \qquad (3)$$

On dit que les événements {A=a} et {B=b} sont statistiquement indépendants si $p_{AB}(a,b) = p_A(a)p_B(b)$. Lorsque cette égalité est vraie pour tout couple AB, alors les espaces A et B sont dits statistiquement indépendants.

1.1.3 Incertitude et information

En suivant le modèle probabiliste, fournir une information à un utilisateur consiste à choisir un événement parmi plusieurs possibles. Qualitativement, fournir une information consiste donc à lever une incertitude sur l'issue d'une expérience aléatoire.

On notera I(a) l'incertitude sur a, encore appelée information propre de a:

$$I(a) = -\log_2 p(a) \qquad (4)$$

Ainsi l'information «b est réalisé» diminue l'incertitude sur a de la quantité:

$$I(a) - I(a \mid b) = \log_2 \frac{p(a \mid b)}{p(a)} \quad (5)$$

Cette dernière quantité est appelée information mutuelle de a et b. Le choix de la fonction log n'est cependant pas arbitraire (D1: expliquer).

1.1.4 Information mutuelle et information propre

On considère un espace probabilisé joint AB où $A = \{a_1,...,a_n\}$ et $B = \{b_1,...,b_m\}$. L'information mutuelle entre les événements {A = a} et {B = b} est définie par:

$$I(a,b) = \log_2 \frac{p(a \mid b)}{p(a)} \quad (6)$$

Par définition p(a,b) = p(a|b)p(b) = p(b|a)p(a). Donc:

$$I(a,b) = I(b,a) = \log_2 \frac{p(a,b)}{p(a)\,p(b)} \qquad (7)$$

Nous allons discuter le signe de I(a; b).

– I(a,b) > 0 signifie que si l'un des deux événements se réalise, alors la probabilité de l'autre augmente.

– I(a,b) < 0 signifie que si l'un des deux événements se réalise, alors la probabilité de l'autre diminue.

– I(a,b) = 0 signifie que les deux événements sont statistiquement indépendants.

L'information propre de l'événement {A=a} est $I(a) = -\log_2 p(a)$. L'information propre s'interprète comme la quantité d'information fournie par la réalisation de cet événement. Notons que l'information propre est toujours positive ou nulle et que, plus un événement est

improbable, plus son information propre est grande. À l'inverse, la réalisation d'un événement certain n'apporte aucune information, ce qui semble conforme à l'intuition.

On peut également définir dans l'espace probabilisé joint AB l'information propre conditionnelle de a sachant b qui est la quantité d'information fournie par l'événement {A=a} sachant que l'événement {B=b} est réalisé:

$$I(a \mid b) = -\log_2 p(a \mid b) \qquad (8)$$

L'information mutuelle entre deux événements est donc:

$$I(a,b) = I(a) - I(a \mid b) \qquad (9)$$

1.1.5 Information mutuelle moyenne et l'entropie

L'information mutuelle moyenne de A et B dans l'espace probabilisé joint AB est définie par:

$$I(A,B) = \sum_{a \in A, b \in B} p(a,b) I(a,b) \quad (10)$$

On peut également définir la moyenne de l'information propre d'un espace probabilisé A. Cette moyenne s'appelle entropie de l'espace A:

$$H(A) = \sum_{a \in A} p(a) I(a) = -\sum_{a \in A} p(a) \log_2 p(a) \quad (11)$$

L'entropie conditionnelle de A sachant B:

$$H(A \mid \mathrm{B}) = -\sum_{a \in A} p(a, \mathrm{b}) \log_2 p(a \mid \mathrm{b}) \quad (12)$$

On en déduit:

$$I(A,B) = H(A) - H(A \mid \mathrm{B}) \quad (13)$$

Propriétés:

1- L'entropie d'une source discrète est maximale lorsque tous les symboles émis par la source sont équiprobables (D2: Démontrer).

2- Soit AB un espace probabilisé joint. L'information mutuelle moyenne H(A,B) de A et B est toujours positive ou nulle. Elle est nulle si A et B sont statistiquement indépendants. Ce résultat signifie essentiellement que, en moyenne, le fait de connaître la valeur de b dans B diminue toujours l'incertitude sur A, sauf si A et B sont indépendants auquel cas aucune information n'est apportée.

2. Codage de Shannon-Fano

La compression probabiliste repose sur le fait qu'un flux de données peut être compacté en utilisant un nombre variable de bit pour représenter les différents octets ou séquences d'octets. Les symboles les plus fréquents sont codés sur de courtes séquences de

bits tandis que les symboles les plus rares sont codés sur davantage de bits. Dans la méthode connue sous le nom de "Shannon-Fano", l'idée est de répartir les symboles en deux groupes de valeur à peu près équivalente, cette valeur étant la somme, dans chaque groupe, des probabilités d'apparition des symboles qu'il contient. Le groupe de gauche est appelé 0, celui de droite 1 (ce choix est arbitraire). Les groupes sont à nouveau subdivisés et nommés 0 ou 1 jusqu'à ce que la subdivision ne contienne plus qu'un symbole. L'arbre binaire ainsi obtenu est formé de segments ou branches et de feuilles. Chaque branche représente un bit d'informations (0 ou 1), Chaque feuille contient un caractère simple. Pour déterminer le code numérique d'un caractère donné, il faut partir du sommet de l'arbre et suivre les branches jusqu'à atteindre la feuille qui le représente. Les caractères les plus fréquents se trouvent le plus près du sommet et requièrent donc moins de bits dans leurs transcriptions compressées.

Exemple:

Soit le message "BANANES ET ANANAS". Les propriétés statistiques de ce message sont données par le tableau ci-dessous. On peut constater qu'un codage optimal doit permettre de coder le message en exactement 43.945 bit, contre 136 en ASCII. D'autre part, on doit être en mesure de coder le symbole A avec moins de 2 bit (1.765 bit très exactement), à condition de coder le symbole T avec 4.08 bit.

s	f(s)	p(s)	H(s)	H	ASCII
A	5	0.294	1.765	8.825	40
N	4	0.235	2.09	8.36	32
E	2	0.118	3.1	6.2	16
S	2	0.118	3.1	6.2	16
<space>	2	0.118	3.1	6.2	16
B	1	0.059	4.08	4.08	8
T	1	0.059	4.08	4.08	8
Total	17		21.315	43.945	136

s: symbole.

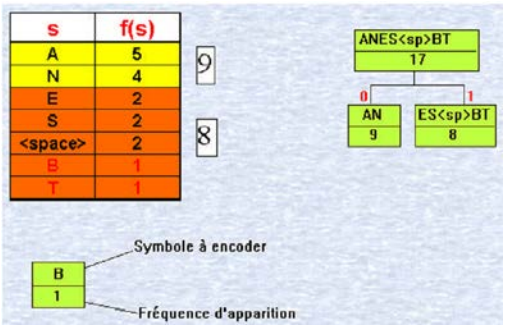
f(s): fréquences d'apparition du symbole.

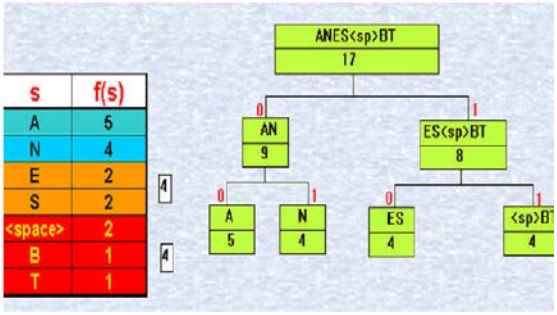
p(s): probabilité d'apparition du symbole.

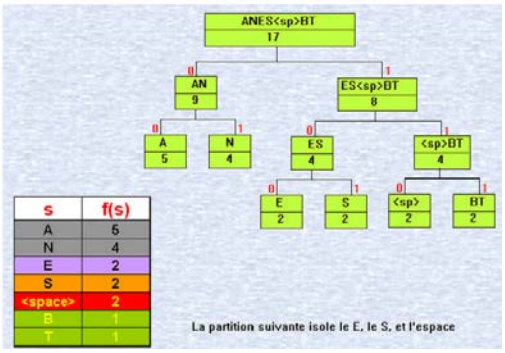
H(s): quantité d'information convoyée par le symbole.

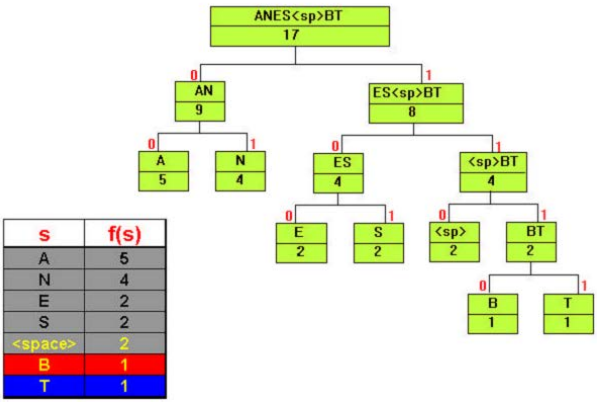
H: quantité d'information cumulée par ce symbole (f(s) * H(s)).

ASCII: quantité de décision selon un code ASCII.









Pour connaître le code associé à chaque lettre, on parcourt l'arbre final de haut en bas, et l'on obtient:

A	00
N	01
E	100
S	101
<space>	110
B	1110
T	1111

On peut s'assurer que le résultat correspond à une quantité de décision très proche de la quantité d'information, soit 44 bit. La redondance résultante est pratiquement nulle.

4. Codage de Huffman

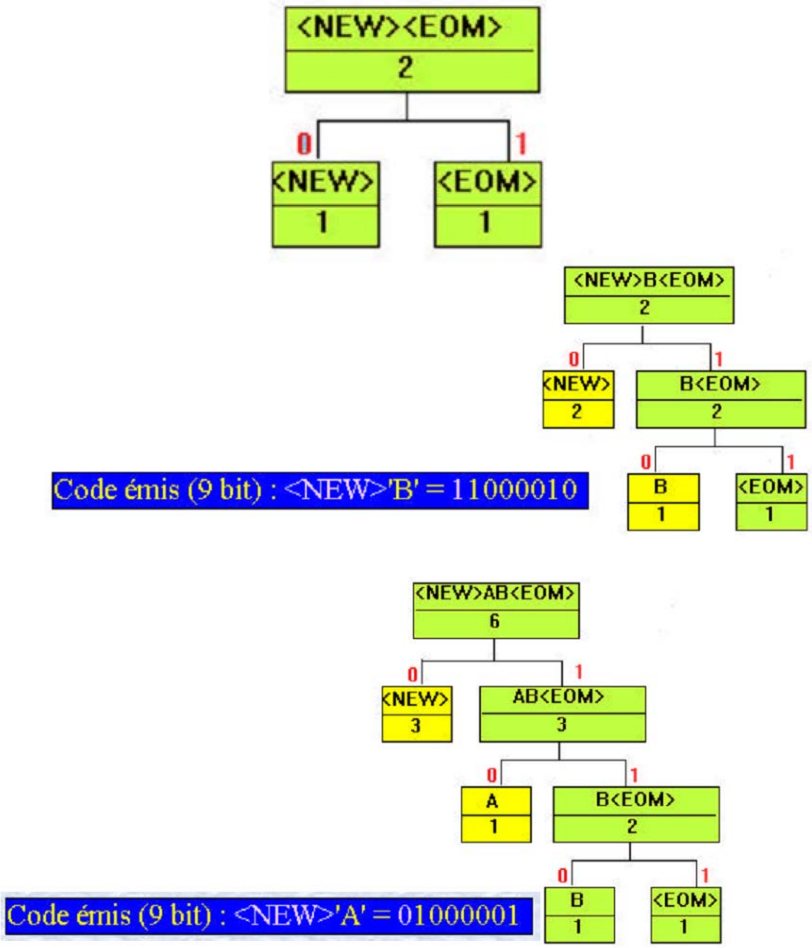
Son étude se fonde sur l'idée que certains caractères sont susceptibles d'apparaître plus souvent que d'autres dans un fichier, laissant la possibilité de les coder sur un nombre de bits plus restreint. Mais il inverse le raisonnement de Shannon et Fano. Ainsi plutôt que de subdiviser une liste indéfiniment avec pour effet de développer un arbre à partir de son sommet, pourquoi ne pas le construire en partant d'en bas ?

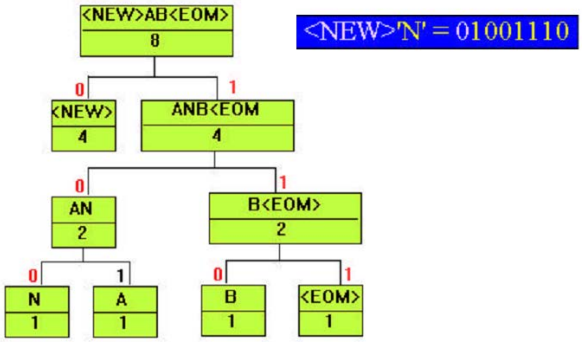
5. Les versions adaptatives de Huffman et Shannon-Fano

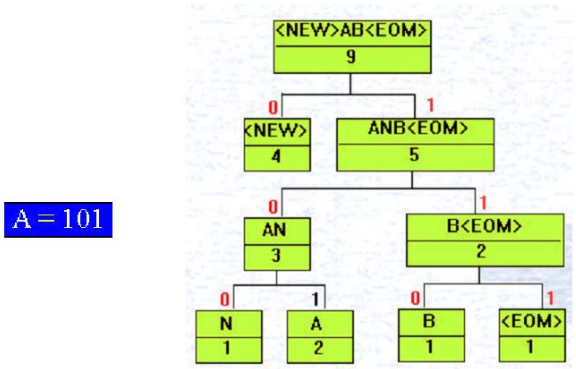
Les codages à base statistique requièrent malheureusement un examen préalable du fichier. Ceci est souvent impossible à réaliser en télécommunications, où l'on désire implémenter, idéalement, la compression dans un modem. Tant les algorithmes de Huffmann que de Shannon-Fano se prêtent à des version adaptatives, permettant le codage en temps réel de l'information à transmettre.

5.1 Codage de Huffman adaptatif

L'idée est de reconstruire l'arbre de codage à chaque nouveau caractère reçu. L'inconvénient est que l'on ne peut pas savoir au préalable de quels symboles va se composer le message, et qu'il est donc nécessaire de pouvoir ajouter en cours de compression de nouveaux symboles, et donc de nouveaux codes, à l'arbre de codage. On débutera l'algorithme avec une table de symboles pratiquement vide, à l'exception de deux symboles particuliers qui sont <NEW> et <EOF>. Le premier sera utilisé pour la définition dynamique de nouveaux symboles, le second pour marquer la fin du message. Ces deux symboles reçoivent arbitrairement, au départ, un "poids" de 1. A chaque fois que l'on rencontre un symbole non encore présent dans l'arbre, on va le signaler en émettant un symbole <NEW> suivi de la définition du symbole (par exemple, sa représentation non compressée en code ASCII ou UNICODE). Ce nouveau symbole sera ensuite inséré dans l'arbre de codage.







6. Codage arithmétique

La technique de codage arithmétique n'a pas, contrairement à celle de Huffman, pour restriction de ne devoir traduire les probabilités que par des nombres entiers de bits. Ce codage traite l'ensemble d'un message comme une seule entité. Il fonctionne par la représentation d'un nombre par un intervalle de nombres réels compris entre 0 et 1.

Exemple

Soit, par exemple, à coder le message “BILL GATES”. Sa distribution de probabilités a l’allure suivante:

SPACE	0.1
A	0.1
B	0.1
E	0.1
G	0.1
I	0.1
L	0.2
S	0.1
T	0.1

On va maintenant associer à chaque symbole un domaine r à l’intérieur de l’espace des probabilités compris entre 0 et 1.

SPACE	0.1	0.00 <= r < 0.10
A	0.1	0.10 <= r < 0.20
B	0.1	0.20 <= r < 0.30
E	0.1	0.30 <= r < 0.40
G	0.1	0.40 <= r < 0.50
I	0.1	0.50 <= r < 0.60
L	0.2	0.60 <= r < 0.80
S	0.1	0.80 <= r < 0.90
T	0.1	0.90 <= r < 1.00

Le premier caractère, 'B', se voit assigner un domaine entre 0.20 et 0.30. Le message final aura donc une valeur comprise entre 0.2 et 0.3, ce qui devient notre nouvel espace de probabilités pour le message. Le caractère I, qui obtient le domaine de 0.5 à 0.6, va utiliser le domaine compris entre 50% et 60% du nouvel espace de probabilités, ce qui amène le message à un nouveau domaine compris entre 0.25 et 0.26.

Symbole	low	high
	0.0	1.00
B	0.2	0.3
I	0.25	0.26
L	0.256	0.258
L	0.2572	0.2576
<Space>	0.25720	0.25724
G	0.257216	0.257220
A	0.2572164	0.2572168
T	0.25721676	0.2572168
E	0.257216772	0.257216776
S	**0.2572167752**	0.2572167756

0.2572167752 est la représentation arithmétique du message "BILL GATES". Le décodage (D3: expliquer) est relativement simple : sachant que le message encodé se trouve entre les bornes 0.2 et 0.3, on voit immédiatement que le premier caractère est B. On soustrait ensuite 0.2 (la borne inférieure) au message, ce qui donne 0.0572167752, ce que l'on divise par l'intervalle de probabilité du symbole B, soit 0.1 : le résultat est 0.572167752, ce qui correspond à l'intervalle du symbole I, et ainsi de suite.

7. Codage LZW (Lempel Ziv Welch)

Le codage de Lempel Ziv Welch est une adaptation du codage LZ78 d'Abraham Lempel et Jacob Ziv amélioré par Terry Welch. Les codages Lempel Ziv ont inventé et posé les bases du codage par dictionnaire et le codage LZW en est donc un lui aussi. Un codage par dictionnaire (ou codage par facteur) est un codage basé sur un dictionnaire ou une liste de mot. Ici, l'objectif va être de remplacer un mot par sa position dans le dictionnaire. Le codage LZW est utilisé dans les format GIF, TIFF et MOD.

1. Introduction

Dans le codage de source, aucun élément extérieur ne vient modifier l'information. Au contraire lorsque l'information transite dans un canal, elle est perturbée par un bruit. Le résultat principal de cette section est qu'il est possible de coder de l'information de façon à ce que la détérioration soit négligeable. Ceci se fait au prix d'une redondance de l'information ou encore d'une vitesse de transmission ou de capacités de stockage plus faibles.

1.1 Définitions

Le canal discret est le plus simple des canaux de transmission. L'ensemble des entrées et l'ensemble des sorties sont deux alphabets finis X et Y. Le canal est discret sans mémoire si le bruit se modélise comme une probabilité conditionnelle de Y sachant X indépendante du temps. Un canal discret sans mémoire est donc défini par:

– l'alphabet d'entrée X.

– l'alphabet de sortie Y.

– la matrice de transition P = (P_{ij}) où $P_{ij} = p(y_j \mid x_i)$.

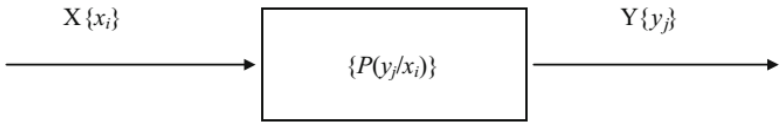


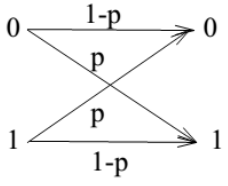
Fig. 1: Canal discret sans mémoire

$$\boldsymbol{P} \equiv [P_{ij}] = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1s} \\ P_{21} & P_{22} & \dots & P_{2s} \\ \vdots & \vdots & \dots & \vdots \\ P_{r1} & P_{r2} & \dots & P_{rs} \end{bmatrix}$$

L'indice i correspond aux lignes (symboles d'entrée) et l'indice j correspond aux colonnes (symboles de sorties).

Par exemple, le canal binaire symétrique de probabilité d'erreur p avec des entrées notées a_1 et a_2 et des sorties notées b_1 et b_2, est défini par les probabilités conditionnelles données par la matrice de transition P :

$$P = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$



Considérons un canal discret et sans mémoire décrit par la matrice de transition P (dimension $r \times s$). Ses éléments sont $p(y_j \mid \mathrm{x_i})$ appelés également les probabilités de transitions. Les probabilités des symboles reçus $p(y_j)$ (j = 1,....s) sont données en fonction de celles des symboles émis $p(x_i)$ (i = 1,....r) telles que:

$$p(y_j) = \sum_{i=1}^{r} p(x_i)\, p(y_j \mid \mathrm{x_i}) \qquad (1)$$

Sachant: $\sum p(y_j) = 1$ et $\sum p(x_i) = 1$.

$p(x_i)$ sont les probabilités d'entrée a priori, $p(y_j \mid \mathrm{x_i})$ sont les probabilités de sortie a posteriori et $p(y_j)$ sont les probabilités de sortie a priori. les probabilités d'entrée a posteriori sont données par:

$$p(\mathrm{x_i} \mid y_j) = \frac{p(x_i)\, p(y_j \mid \mathrm{x_i})}{p(y_j)} \qquad (2)$$

La probabilité conjointe est données par (formule de Bayes):

$$p(\mathrm{x_i}, y_j) = p(x_i)\, p(y_j \mid \mathrm{x_i}) = p(y_j)\, p(\mathrm{x_i} \mid y_j) \quad (3)$$

En utilisant la formule (1), la formule (2) devient:

$$p(\mathrm{x_i} \mid y_j) = \frac{p(x_i)\, p(y_j \mid \mathrm{x_i})}{\sum_{i=1}^{r} p(x_i)\, p(y_j \mid \mathrm{x_i})} \qquad (3)$$

Sachant: $\sum_i p(x_i \mid \mathrm{y_j}) = 1$ et $\sum_i \sum_j p(x_i, y_j) = 1$.

L'entropie a priori est donnée par:

$$H(X) = \sum_{i=1}^{r} p(x_i) \log_2 \frac{1}{p(x_i)} \qquad (4)$$

Après la réception d'un symbole $\mathrm{y_j}$, l'entropie a posteriori partielle de l'entrée (l'ensemble X) est:

$$H(X \mid \mathrm{y_j}) = \sum_{i=1}^{r} p(x_i \mid \mathrm{y_j}) \log_2 \frac{1}{p(x_i \mid \mathrm{y_j})} \qquad (5)$$

L'entropie a posteriori de l'entrée (l'ensemble X)

$$H(X \mid \mathrm{Y}) = \sum_{i=1}^{r} \sum_{j=1}^{s} p(x_i, \mathrm{y_j}) \log_2 \frac{1}{p(x_i \mid \mathrm{y_j})} \qquad (6)$$

Soit l'émission d'un symbole $\mathrm{x_i}$ à travers un canal. l'incertitude sur ce symbole est $\log_2(1/p(\mathrm{x_i}))$. Si le symbole reçu est $\mathrm{y_j}$, l'incertitude sur le symbole $\mathrm{x_i}$ est $\log_2(1/p(\mathrm{x_i}/\mathrm{y_j}))$. La

quantité d'informations transmises et, alors, la différence entre l'incertitude a priori et celle a posteriori telle que:

$$I(x_i, y_j) = \log_2(\frac{1}{p(x_i)}) - \log_2(\frac{1}{p(x_i \mid \mathrm{y_j})}) \quad (7)$$

La quantité moyenne d'informations transmises (information mutuelle moyenne) est alors:

$$I(X,Y) = H(X) - H(X \mid Y) \qquad (8)$$

On peut mettre:

$$I(X,Y) = \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i, y_j)\log_2\left(\frac{p(x_i, y_j)}{p(x_i)p(y_j)}\right) \quad (9)$$

L'entropie conjointe de X et Y est:

$$H(X,Y) = \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i, y_j)\log_2\left(\frac{1}{p(x_i, y_j)}\right) \quad (10)$$

L'entropie des symboles reçus est:

$$H(Y) = \sum_{j=1}^{s} p(y_j)\log_2\left(\frac{1}{p(y_j)}\right) \quad (11)$$

L'entropie conditionnelle des symboles reçus est

$$H(Y \mid \mathrm{X}) = \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i, y_j)\log_2\left(\frac{1}{p(y_j \mid \mathrm{x_i})}\right) \qquad (12)$$

On peut vérifier que:

$$H(X,Y) = H(X) + H(Y) - H(X \mid \mathrm{Y}) = H(X) + H(\mathrm{Y} \mid \mathrm{X}) = H(Y) + H(X \mid \mathrm{Y}) \qquad (13)$$

Propriétés:

- Un canal est dit sans perte si H(Y|X) = 0 pour toutes les distributions en entrée.
- Un canal est dit déterministe si $p(y_j|x_i)$ = 0 ou 1 pour tout i,j. Ou encore si H(Y|X) = 0. La sortie de canal est déterminée par l'entrée.
- Un canal est dit sans bruit s'il est déterministe et sans perte.
- Un canal est inutile si H(X,Y) = 0 pour toutes les toutes les distributions en entrée.
- Un canal est symétrique si chaque ligne de Π contient (à permutation près) les mêmes valeurs $p_1,...p_m$ et chaque colonne de Π contient (à permutation près) les mêmes valeurs $q_1,...q_n$.

1.2 Capacité d'un canal

L'information mutuelle moyenne peut être décrite en fonction des probabilités a priori (caractéristiques de la source) et la matrice de transition (caractéristiques du canal) telle que:

$$I(X,Y) = \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i)\,p(y_j \mid \mathrm{x_i}) \log_2 \left(\frac{p(y_j \mid \mathrm{x_i})}{\sum_{i=1}^{r} p(x_i)\,p(y_j \mid \mathrm{x_i})} \right) \quad (14)$$

On remarque que l'information mutuelle dépend de la manière dont le canal transmet le information ainsi que la source à l'entrée du canal. L'information mutuelle maximale est donnée par:

$$I_{\max} = \max I(X,Y) \qquad (15) \qquad \text{(Sh/symb)}$$

Pour un débit symbole donné $v_m(X,Y)$, le débit d'information maximal (débit maximal du canal) est donné par:

$$C = v_m(X,Y) I_{\max} \qquad (16)$$

la capacité telle qu'elle est définie ci-dessus est égale au plus grand nombre de bits d'information qui peuvent être transmis sur le canal avec un taux d'erreurs aussi faible que possible.

Dans le cas d'un canal binaire symétrique, la capacité est donnée par:

$$C = 1 + p \log_2 p + (1-p)\log_2 (1-p) \quad (17)$$

Dans le cas d'un canal continu dont le bruit est gaussien et additif, la capacité est donnée par:

$$C = B \log_2 (1 + \frac{S}{N}) \qquad (18)$$

avec: B la bande passante du canal et S/N est le rapport signal bruit.

2. Codage canal

Lorsque l'entropie de la source est égale à $H_{max}(X)$, H(X|Y) ne dépend plus que du canal de transmission utilisé. Si H(X|Y) est non négligeable (cas du canal bruyant), il ne sera pas possible d'effectuer une communication sans erreur en reliant directement la source au canal de transmission. Il faut donc placer un élément appelé codeur de canal entre la source et le canal de transmission (figure ci-dessous).

Fig. 1: système de communication avec codage de canal.

Pour ce nouveau système, on peut définir l'information mutuelle moyenne I(U, V) = H(U) − H(U|V). Le rôle du codage de canal est de rendre la quantité d'information moyenne H(U|V) aussi faible que souhaité. Il est alors possible de transmettre au travers de ce canal bruité une quantité d'information moyenne H(U) avec le critère de qualité souhaité.

2.1 Théorème fondamental du codage de canal

Il existe un codage de canal permettant de garantir une communication avec un taux d'erreurs aussi faible que souhaité à la condition que la quantité d'information moyenne entrant dans l'ensemble codeur-canal-décodeur soit inférieure à la capacité C du canal:

$$H(U) < C \qquad (19)$$

3. Codes correcteurs d'erreur

L'objectif du codage de canal est de protéger les données issues du codage de source contre les erreurs de transmission. Une première solution consiste à utiliser un codage aléatoire puisque ce codage permet d'atteindre la limite du théorème du codage de canal. Soit un code C en bloc binaire aléatoire (N, K) comprenant 2^K mots de code. Chaque mot d'information composé de K bits est associé à un mot de code unique composé de N bits. Pour réaliser un codeur aléatoire, il faut tout d'abord construire une liste de 2^K mots de code. Chacun des mots de code est composé de N bits tirés aléatoirement. L'opération de codage consiste à associer à chaque mot d'information une adresse unique qui servira ensuite pour lire le mot de code correspondant. Le contenu de la liste ne présentant aucune structure particulière, l'opération de décodage implique de faire la comparaison exhaustive du mot reçu en sortie du canal avec l'ensemble des 2^K mots de code avant de déterminer le mot de code le plus probable. La complexité du décodeur croit exponentionellement avec K et rend cette technique presque toujours inutilisable en pratique. L'impossibilité pratique d'utiliser le codage aléatoire nous amène donc à utiliser des codes possédant une structure algébrique comme par exemple la linéarité et rendant ainsi les opérations de codage et de décodage plus simples à effectuer.

Nous nous intéresserons aux trois principales familles de codes suivantes:

1- les codes en bloc linéaires

2- les codes convolutifs

3- les codes concaténés

3.1 Les corps finis

Un corps F est un ensemble non vide muni de deux lois de composition internes, l'addition et la multiplication et satisfaisant les axiomes suivants:

1- F est un groupe commutatif par rapport à l'addition (associativité, élément neutre noté 0, symétrique, commutativité).

2- la multiplication est associative: si a, b, c $\in$ F, alors a(bc) = (ab)c.

3- la multiplication est commutative: si a, b $\in$ F, alors ab = ba.

4- la multiplication est distributive à droite et à gauche par rapport à l'addition : si a, b, c $\in$ F, alors a(b + c) = ab + ac et (a + b)c = ac + bc.

5- le corps contient un élément neutre noté 1 pour la multiplication.

6- tout élément de F non nul est inversible ; si a $\in$ F($a \neq 0$), a^{-1} est son inverse avec $aa^{-1} = 1$.

3.2 Les corps de Galois

Un corps de Galois est un corps fini possédant q éléments. Il est noté GF(q) (*Galois Field* en anglais) ou F_q. Il est possible de construire un corps de Galois à condition que q soit un nombre premier ou soit de la forme $q = p^m$ avec p nombre premier. Tout corps de Galois doit contenir au moins les éléments neutres 0 et 1. Ainsi, le corps de Galois le plus simple est le corps GF(2).

- Addition et multiplication dans GF(2):

+	0	1
0	0	1
1	1	0

*	0	1
0	0	0
1	0	1

3.3 Codes en bloc linéaires binaires

Un code C en bloc linéaire q-aire (N, K) est un ensemble comprenant q^K mots de code. On associe à chaque mot d'information composé de K symboles q-aire un mot de code composé de N symboles q-aire. Un code en bloc est linéaire si les N symboles du mot code sont obtenus par combinaison linéaire des K symboles du mot d'information. Cette propriété permet en particulier de décrire l'opération de codage sous une forme matricielle. Dans la suite, nous nous intéresserons aux codes en bloc linéaires binaires pour lesquels on a q = 2.

Il est pratique de représenter les mots d'information et les mots de code par des vecteurs. Soit **u** = [u_1, u_2, . . . , u_K] un mot d'information composé de K bits d'information et **c** = [c_1, c_2, . . ., c_N] le mot de code associé composé de N bits. On a la relation matricielle suivante entre le mot d'information **u** et le mot de code associé **c**:

$$c = uG \quad (20)$$

G est la matrice génératrice du codeur de dimension K × N.

$$\mathbf{G} = \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_K \end{pmatrix} = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ g_{K1} & g_{K2} & \cdots & g_{KN} \end{pmatrix} \quad (21)$$

Chaque mot de code est une combinaison linéaire des vecteurs $\mathbf{g}_i$ de **G**.

Il est toujours possible en combinant les lignes entre elles de mettre la matrice génératrice G sous la forme systématique suivante:

$$\mathbf{G} = [\mathbf{I_K} \quad \mathbf{P}] = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & p_{11} & p_{12} & \dots & p_{1N-K} \\ 0 & 1 & 0 & \dots & 0 & p_{21} & p_{22} & \dots & p_{2N-K} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 & p_{K1} & p_{K2} & \dots & p_{KN-K} \end{pmatrix} \quad (22)$$

Exemple: code C_3 (7, 4) avec la matrice génératrice suivante :

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Ici, la matrice génératrice sous la forme systématique s'obtient simplement en ajoutant les lignes 3 et 4 à la ligne 1, et la ligne 4 à la ligne 2. Cette technique de combinaison de ligne permet toujours de convertir une matrice génératrice quelconque en une matrice génératrice systématique. On peut vérifier que la forme systématique de cette matrice génératrice est la suivante:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Les trois bits de parité sont obtenus comme suit:

$$c_5 = u_1 + u_2 + u_3$$
$$c_6 = u_2 + u_3 + u_4$$
$$c_7 = u_1 + u_2 + u_4$$

3.3.1 Propriétés et définitions

1- le rendement R d'un code en bloc (N, K) est égal à :

$$R = K / N \qquad (22)$$

2- soit $\mathbf{c}_1$ et $\mathbf{c}_2$ deux mots de code du code C, et α_1 et α_2 deux éléments du corps fini. La linéarité implique que $\alpha_1\mathbf{c}_1 + \alpha_2\mathbf{c}_2$ est aussi un mot de code de C. Par conséquence, le mot $\mathbf{c}_0$ = [00 . . . 0] est toujours un mot de code d'un code linéaire. On appellera ce mot de code le mot de code nul.

3- soit $\mathbf{c}_1$ et $\mathbf{c}_2$ deux mots de code du code C de longueur N, la distance de Hamming $d_H(\mathbf{c}_1, \mathbf{c}_2)$ est égale aux nombres de bits qui différent.

Exemple: $\mathbf{c}_1$ = [001100] et $\mathbf{c}_2$ = [001111], $d_H(\mathbf{c}_1, \mathbf{c}_2) = 2$.

4- le poids de Hamming $w(\mathbf{c})$ d'un mot de code binaire **c** est égal au nombre de bits non nuls de ce mot de code.

5- La distance minimale d_{min} du code C est le nombre de bits qui différent entre les deux mots de code les plus proches au sens de la distance de Hamming: $d_{min} = \min_{i,j,i\neq j} d_H(\mathbf{c}_i, \mathbf{c}_j)$

6- Lorsque le code est linéaire, la distance minimale d_{min} est égale au poids de Hamming minimal du code C (en excluant le mot de code nul $\mathbf{c}_0$).

3.3.2 Matrice de contrôle

Associé à chaque code C linéaire en bloc binaire (N, K) il existe un code linéaire en bloc binaire dual (N, N-K). Soit **H** la matrice génératrice de ce code dual. Chacun des mots de code **c** du code C est orthogonal à tous les mots de code du code dual:

$$cH^T = 0 \qquad (23)$$

Puisque cette relation est valide pour tous les mots de code du code C, on a la relation entre la matrice génératrice **G** du code C et **H**:

$$GH^T = 0 \qquad (24)$$

Si la matrice génératrice **G** est systématique, **H** est de la forme suivante:

$$\mathbf{H} = [\mathbf{P}^T \quad \mathbf{I}_{N-K}] = \begin{pmatrix} p_{11} & p_{21} & \dots & p_{K1} & 1 & 0 & 0 & \dots & 0 \\ p_{12} & p_{22} & \dots & p_{K2} & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{1N-K} & p_{2N-K} & \dots & p_{KN-K} & 0 & 0 & 0 & \dots & 1 \end{pmatrix} \qquad (25)$$

La matrice **H** est appelée matrice de contrôle ou matrice de parité du code C.

Exemple (suite): la matrice de contrôle du code C_3 (7, 4) est la suivante:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

3.4 Décodage des codes en bloc linéaires binaires

Le mot reçu **y** est la somme modulo 2 du mot de code émis **c** et d'un vecteur d'erreurs **e**:

$$\mathbf{y} = \mathbf{c} + \mathbf{e} \qquad (26)$$

En multipliant le mot reçu **y** par la matrice de parité transposée $\mathbf{H}^T$, on obtient le syndrome d'erreurs **s** de dimension $1\times(N - K)$:

$$\begin{aligned}\mathbf{s} &= \mathbf{y}\mathbf{H}^T \\ &= \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T \\ &= \mathbf{e}\mathbf{H}^T \qquad \text{car} \quad \mathbf{c}\mathbf{H}^T = 0\end{aligned}$$

Le décodage par syndrome consiste tout d'abord à calculer le syndrome d'erreurs correspondant au mot reçu. Ensuite, on associe au syndrome le vecteur d'erreurs estimé correspondant $\hat{e}$. Il suffit donc de stocker dans une table de correspondance les syndromes et vecteurs d'erreurs.

$\hat{\mathbf{e}}$	$\mathbf{s}$
$\hat{\mathbf{e}}_0$	$\mathbf{s}_0$
$\hat{\mathbf{e}}_1$	$\mathbf{s}_1$
$\hat{\mathbf{e}}_2$	$\mathbf{s}_2$
$\vdots$	$\vdots$
$\hat{\mathbf{e}}_{2^{N-K}-1}$	$\mathbf{s}_{2^{N-K}-1}$

Exemple:

considérons le code C_4 (5,2) défini par la matrice génératrice suivante:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

et la matrice de parité associée H suivante:

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Considérons le mot d'information **u** = [11]. Le mot de code associé est **c** = [11110]. On peut vérifier que le syndrome associé à ce mot de code est nul: **s** = [000].

Supposons qu'une erreur survienne dans la transmission sur le 4-ième bit du mot émis: **e** = [00010]. Le mot reçu est alors **y** = [11100].

Pour construire la table de syndrome, nous calculons le syndrome correspondant à chaque vecteur d'erreur.

$\hat{\mathbf{e}}$	$\mathbf{s}$
00000	**000**
00001	**001**
00010	**010**
00100	**100**
01000	**011**
10000	**101**
11000	**110**
10010	**111**

Le calcul du syndrome associé au mot reçu **y** donne **s** = **yH**^{T} = [010]. En utilisant cette table de décodage par syndrome on trouve $\hat{e}$ = [00010]. En ajoutant le vecteur d'erreurs estimé $\hat{e}$ au mot reçu **r** on retrouve bien $\hat{e}$ = [11110]. On peut voir que ce code permet de corriger tous les motifs d'erreurs simples.

3.5 Capacité de correction d'erreurs d'un code linéaire binaire en bloc

Le nombre d'erreurs e qu'est capable de corriger un code correcteur d'erreurs dépend de la distance minimale du code. Les 2^K mots de code du code peuvent être vus comme les centres de boules de Hamming de rayon e. Pour garantir que les 2^K sphères ne se chevauchent pas, on doit garantir:

$$e = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \quad (27)$$

3.6 Codes en bloc principaux

3.6.1 Codes de Hamming

Les codes de Hamming sont des codes en bloc linéaires binaires (N, K) avec $N = 2^J - 1$ et $K = 2^J - 1 - J$. Un code de Hamming (N, K) peut être décrit simplement à partir de sa matrice de contrôle **H** de dimension J × N. En effet, les colonnes de **H** sont les N vecteurs binaires non nuls avec J éléments. Par exemple pour J = 3, le code de Hamming est un code (7,4). La matrice de parité est la suivante:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

La distance de Hamming de ces codes est égale à 3 et ils peuvent donc corriger une erreur. Au décodage, une valeur non nulle du syndrome donne directement la position de l'erreur.

3.6.2 Code de Golay

Le code de Golay est un code linéaire binaire (23,12) dont la distance minimale est égale à 7. La matrice génératrice systématique du code de Golay (23,12) est la suivante:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

3.7 Codes cycliques

Les codes cycliques représentent la classe la plus importante des codes en blocs linéaires. Leur mise en œuvre relativement aisée, à partir de registres à décalage et d'opérateurs logiques, en a fait des codes attractifs et très utilisés

3.7.1 Définition et représentation polynomiale

Un code en blocs linéaire $C(n, k)$ est cyclique si pour tout mot de code $c = [c_1 c_2 ... c_n]$, $c_1 = [c_n c_1 ... c_{n-1}]$, obtenu par permutation circulaire à droite d'un symbole de **c**, est aussi un mot de code. Cette définition des codes cycliques entraîne que toute permutation circulaire à droite de j symboles d'un mot de code, redonne un mot de code. Pour les codes cycliques, on utilise une représentation polynomiale des mots de code et des blocs de données. Ainsi, au mot de code **c** on associe le polynôme $c(x)$ de degré $n-1$:

$$\mathrm{c(x)} = c_0 + c_1 x + c_2 x^2 + ... + c_j x^j + c_{n-1} x^{n-1} \qquad (28)$$

et au bloc de données **d** le polynôme $d(x)$ de degré $k-1$:

$$\mathrm{d(x)} = d_0 + d_1 x + d_2 x^2 + ... + d_j x^j + d_{k-1} x^{k-1} \quad (29)$$

où d_j et c_j prennent leurs valeurs dans F_2.

Multiplions $c(x)$ par x:

$$xc(x) = c_0 x + c_1 x^2 + ... + c_j x^{j+1} + c_{n-1} x^n \qquad (30)$$

puis divisons $xc(x)$ par x^n+1, nous obtenons:

$$xc(x) = c_{n-1}(x^n + 1) + c_1(x) \qquad (31)$$

où $c_1(x)$ est le reste de la division de $xc(x)$ par $x^n + 1$ avec:

$$c_1(x) = c_{n-1} + c_0 x + ... + c_j x^{j+1} + c_{n-2} x^{n-1} \qquad (32)$$

On peut noter que $c_1(x)$ correspond au mot de code $c_1 = [c_{n-1} c_0 c_1 ... c_{n-2}]$.En suivant la même démarche que précédemment, on obtient:

$$x^j c(x) = (x^n + 1) q(x) + c_j(x) \quad (33)$$

où $c_j(x)$ est aussi un mot de code obtenu par j permutations circulaires à droite des symboles de $c(x)$.

Les mots de code d'un code cyclique sont des multiples d'un polynôme $g(x)$ normalisé de degré $(n-k)$ appelé polynôme générateur:

$$c_1(x) = g_0 + g_1 x + ... + g_j x^j + g_{n-k} x^{n-k} \qquad (34)$$

où g_j prend ses valeurs dans F_2.

Le polynôme générateur d'un code cyclique est un diviseur de x^n+1. Il existe un polynôme $h(x)$ de degré k tel que l'équation (35) soit vérifiée:

$$g(x)h(x) = x^n + 1 \qquad (35)$$

Le produit $d(x)g(x)$ est un polynôme de degré inférieur ou égal à $n-1$, il peut en conséquence représenter un mot de code. Le polynôme $d(x)$ possédant 2^k réalisations, $d(x)g(x)$ permet de générer 2^k mots de code. Notons $d_l(x)$ la l-ième réalisation de $d(x)$ et $c_l(x)$ la représentation polynomiale du mot de code associé. Nous pouvons écrire:

$$c_l(x) = d_l(x)g(x) \qquad (36)$$

3.7.2 Matrice génératrice d'un code cyclique

À partir du polynôme générateur $g(x)$ il est possible de construire une matrice génératrice **G** du code $C(n, k)$. Soit $g(x) = g_0 + g_1x + ... + g_tx^t$ le générateur d'un code cyclique C de longueur n. Une matrice génératrice de C est donnée par:

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_t & 0 & \dots & \dots & 0 \\ 0 & g_0 & g_1 & g_2 & \dots & g_t & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & g_0 & g_1 & g_2 & \dots & g_t & 0 \\ 0 & \dots & \dots & 0 & g_0 & g_1 & g_2 & \dots & g_t \end{pmatrix} \qquad (37)$$

3.7.2 Matrice de contrôle

Soit C un code cyclique de générateur $g(x)$. Le polynôme de contrôle de C est définit par:

$$h(x)g(x) = (x^n + 1) \qquad (38)$$

On peut construire la matrice de contrôle H telle que:

$$H = \begin{pmatrix} 0 & \dots & & 0 & h_k & \dots & h_1 & h_0 \\ 0 & \dots & 0 & h_k & \dots & h_1 & h_0 & 0 \\ \vdots & & \ddots & & & \ddots & & \vdots \\ 0 & & & & & & & \\ h_k & \dots & h_1 & h_0 & 0 & \dots & & 0 \end{pmatrix} \qquad (39)$$

1. Introduction

La compression est l'action utilisée pour réduire la taille physique d'un bloc d'information. En compressant des données, on peut placer plus d'informations dans le même espace de stockage, ou utiliser moins de temps pour le transfert au travers d'un réseau téléinformatique. Parce que généralement les images requièrent une place importante, la compression est devenue part intégrante des données graphiques. On rencontre souvent la compression de données comme étant une partie du codage de données au même titre que le cryptage de données (cryptographie) et la transmission de données.

Presque chaque format de fichier incorpore l'une ou l'autre méthode de compression. Parmi les plus connues de ces méthodes, on peut citer: Pixel Packing, RLE, Lempel-Ziv, JPEG, ...etc. Récemment, Google a proposé un nouveau algorithme JPEG pour la compression d'images nommé Guetzli (un terme suisse allemand qui désigne un biscuit).

la problématique de la compression d'image consiste à satisfaire les contraintes technologiques, techniques ou financières auxquelles on est confronté, tout en obtenant la qualité requise de l'image décomprimée pour l'application désirée.

2. Définitions

Les termes données brutes (raw data) et données non codées (unencoded data) désignent les données avant qu'elles ne soient compressées et les termes données codées (encoded data) ou données compressées (compressed data) désignent les données après qu'elles aient été compressées.

Le terme taux de compression (compression ratio) est utilisé pour se référer aux rapport entre la taille des données non compressées sur taille des données compressées. Ce sera également un critère d'efficacité entre différents algorithmes.

Une méthode de compression symétrique utilise le même algorithme, et demande la même capacité de calcul, aussi bien pour la compression que pour la décompression. Les méthodes de compression asymétriques demandent plus de travail dans une direction que dans l'autre.

Une méthode de compression de données est dite sans perte lorsque des données sont compressées et ensuite décompressées, l'information originale contenue dans les données a été préservée. Aucune donnée n'a été perdue ou oubliée. Les données n'ont pas été modifiées. La méthode de compression avec pertes quant à elle "jette", de façon sélective, quelques données d'une image dans le but d'effectuer la compression avec un taux de compression meilleur que la plupart des méthodes de compression sans pertes. Les algorithmes avec pertes s'appliquent généralement aux données ayant de forts taux de redondance, comme les images, ou les sons.

En fin, la compression d'images peut s'effectuer dans deux domaines dont une image peut être représentée de deux façons strictement équivalentes:

1- Dans le domaine spatial, dans lequel l'image est représentée sous forme de pixels. C'est le domaine accessible visuellement par l'observateur.

2- Dans un domaine fréquentiel, dans lequel l'image est représentée sous forme de coefficients de fréquences spatiales.

Le passage d'un domaine à l'autre se fait par des transformations mathématiques totalement réversibles, telles que la transformation en ondelettes ou la transformation Cosinus.

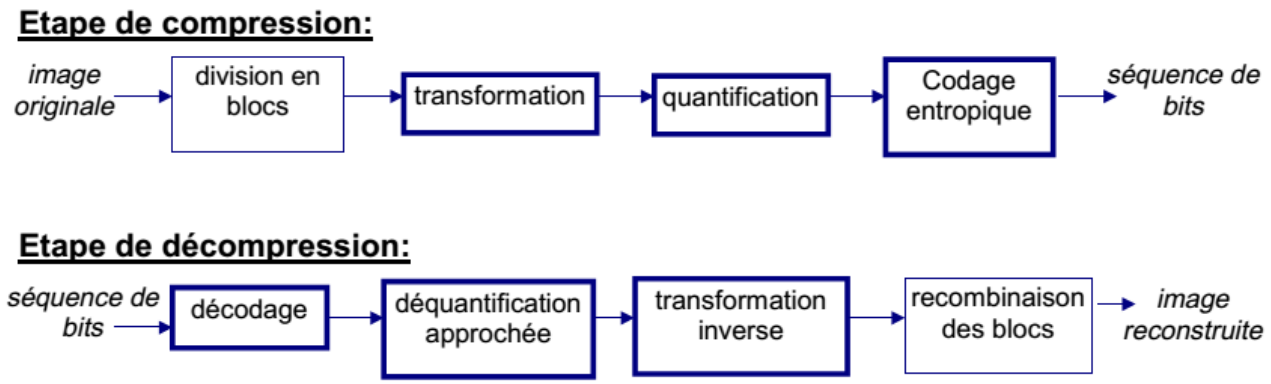
Dans le domaine spatial, l'information contenue dans l'image est distribuée sur toute la matrice image. Dans le domaine des fréquences, l'information (qui est strictement équivalente) est généralement plus "concentrée". De ce fait, il est approprié de construire un algorithme de compression sur les coefficients du plan des fréquences de l'image. Cette approche est largement utilisée, par exemple dans les méthodes standards JPEG et MPEG.

3. Méthodes de compression basées sur les transformations

Les méthodes par transformation figurent parmi les techniques de compression les plus employées. Elles permettent d'obtenir des taux de compression élevés tous en conservant une bonne qualité d'image. Ce sont des méthodes qui font appel successivement à plusieurs principes de compression. Elles sont utilisées par des standards internationaux pour le codage des images fixes et de la vidéo.

Le principe de la compression par transformation est de décomposer les pixels fortement corrélés de l'image en ensembles de coefficients spectraux partiellement décorrélés, dont l'énergie est concentrée dans un nombre restreint de coefficients. Ce compactage de l'énergie permet d'affecter en priorité aux coefficients spectraux les plus énergétiques un nombre de bits plus élevé qu'aux autres.

Les méthodes par transformation suivent le schéma de fonctionnement présenté dans la figure ci-dessous:



3.1 Division en blocs

La première étape consiste à diviser l'image en blocs sur lesquels vont s'appliquer indépendamment les étapes suivantes. La principale raison de ce découpage est de limiter le nombre

de pixels à traiter à la fois pour diminuer les temps de calcul et la complexité des circuits électroniques. La taille des blocs est variable selon les méthodes. Elle est de 8x8 pour JPEG et MPEG.

3.2 Transformation

La deuxième étape consiste à appliquer une transformation mathématique à chaque bloc. Le but de cette transformation est de décorréler les pixels, ce qui a pour effet en général de redistribuer l'énergie de l'image dans un nombre restreint des coefficients transformés. De cette façon, un grand nombre de coefficients transformés ont des très faibles valeurs, et peuvent être supprimés ou se voir allouer un nombre très faible de bits lors de l'étape suivante de quantification.

Il existe des nombreuses transformées orthogonales. On peu citer: La transformée de Fourier discrète (DFT), la transformée discrète en cosinus (DCT) et en sinus (DST), la transformée de Karhunen-loeve (KLT) et la transformée en ondelettes discrète (DWT)

3.3 Quantification et codage

La troisième étape est la quantification des coefficients transformés, afin de se ramener à un nombre limité de valeurs différentes. La quantification est souvent précédée d'une pondération psychovisuelle des coefficients, afin de préserver ceux auxquels l'oeil est le plus sensible. La quantification est la seule étape irréversible de tout le schéma de compression par transformation. Souvent, un quantificateur scalaire uniforme est employé. Un codage entropique est effectué sur les coefficients quantifiés, pour aboutir au flot binaire de sortie.

4. Mesures de performances

Afin d'évaluer les performances d'une méthode de compression d'images, plusieurs critères objectifs sont proposés.

4.1 Taux de compression

IL est définit par:

TC=Nombre de bits de l'image originale/ Nombre de bits de l'image comprimée.

Pour une même méthode de compression et un même TC réalisés sur des images distinctes, la qualité obtenue peut être très variable d'une image à l'autre. Les propriétés statistiques des images originales jouent un rôle prépondérant dans le résultat obtenu.

4.2 Mesure de la distorsion

La distorsion (D) est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement identique à l'image originale. La mesure de distorsion utilisée généralement en compression d'image est l'erreur quadratique

moyenne MSE. Cette grandeur est définie par la moyenne des écarts au carré e_{mn}^2 entre le pixel (m,n) de l'image originale I(m,n), et le pixel (m,n) de l'image reconstruite Î(m,n):

$$MSE = \frac{1}{M.N}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[I(m,n) - \hat{I}(m,n)]^2$$

4.3 Rapport signal/bruit

Le PSNR est définit en fonction de MSE tel que:

$$PSNR \text{ ou PPSNR} = 10.\log_{10}.\frac{(2^R-1)^2}{MSE} \quad \text{dB}$$

R étant le nombre de bits par pixel (débit).

L'inconvénient de la MSE est qu'elle ne rend pas compte de la perte de qualité visuelle engendrée par la compression. Si tous les pixels d'une image étaient translatés, l'erreur quadratique serait très élevée, alors que la qualité visuelle serait parfaitement bonne. De plus. la MSE est une mesure globale sur toute l'image, qui gomme les variations locales. Par exemple dans une image médicale, si des détails anatomiques importants sont dégradés par la compression et si la majeure partie du reste de l'image est fidèlement restituée, alors la MSE est relativement faible mais, pour l'expert médical, cette image a une qualité diagnostique médiocre.

5. Normes et organismes de normalisation

Il existe plusieurs formats d'image parmi lesquels on trouve: JPEG, JBIG, H-261 et MPEG.

5.1 JPEG

Dans la fin des années 80, le comité Joint Photographic Experts Group (JPEG) a sélectionné en tant que standard mondial pour le codage des images fixes en couleurs une méthode de compression basée sur un schéma par transformation DCT. Le format d'images JPEG est très adapté aux photographies ou images volumineuses qui contiennent des millions de couleurs telles qu'un fond d'écran, l'image d'une famille, capture d'écran vidéo etc. La compression JPEG entraîne une perte de données irréversible car l'algorithme de compression utilisé est destructif, ce qui peut aboutir à une perte de qualité plus ou moins perceptible par l'œil humain en fonction du taux de compression utilisé.

5.1 JBIG

Destinée à la compression des images ne comptant que deux tons de couleur (bi-level images, par exemple les images de fax). JBIG publiée par l'ISO et développée initialement par le *Joint Bi-level Image experts Group* est regroupé aujourd'hui avec JPEG.

5.3 H-261

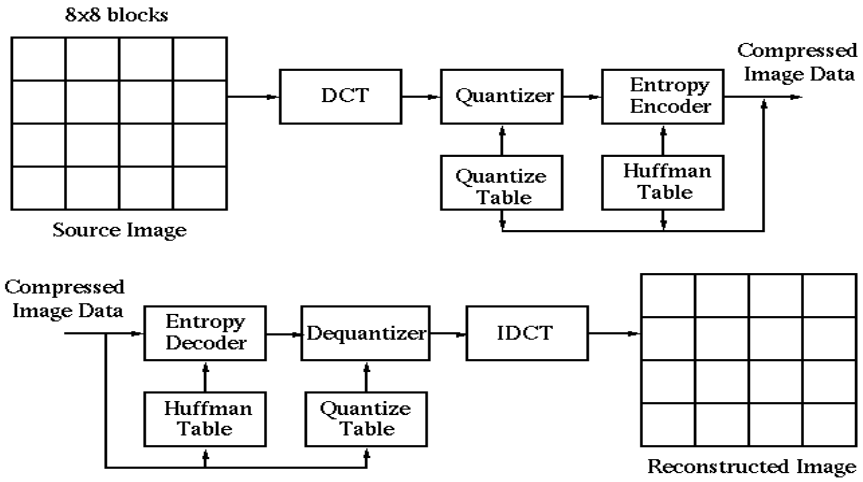
Le format H-261 a été développé par l'UIT. C'est un standard destinée à la compression des images animées pour la visiophonie. Le H-61 utilise un codage hybride combinat la DCT et le codage prédictif.

5.3 MPEG

Le groupe de travail Moving Pictures Experts Group (MPEG) a publié au début des années 90 un standard pour la vidéo qui comprend le codage des séquences d'images couleur et du son associe. La méthode MPEG tire profit de la ressemblance des images successives dans une séquence d'image. La première hypothèse de base est que chaque image contient des objets en mouvement, dont la position peut être prédite par leur position dans l'image précédente par simple translation des pixels. Dans la méthode MPEG, l'image est découpée en zone de 16x16 pixels, les objets, dont on estime le mouvement. La deuxième hypothèse est que l'intensité lumineuse d'un objet en mouvement reste la même d'une image à l'autre.

6. Principe de la compression JPEG

Le principe de l'algorithme JPEG pour une image à niveaux de gris est le suivant: Une image est décomposée séquentiellement en blocs de 8x8 pixels subissant le même traitement. Une transformée en cosinus discrète bidimensionnelle est réalisée sur chaque bloc. Les coefficients de la transformée sont ensuite quantifiés uniformément en association avec une table de 64 éléments définissant les pas de quantification. Une table type est fournie par le standard mais n'est pas imposée. Un codage entropique, sans distorsion, est enfin réalisé permettant d'utiliser les propriétés statistiques des images. On commence par ordonner les coefficients suivant un balayage en zigzag pour placer d'abord les coefficients correspondant aux fréquences les plus basses. Cela donne une suite de symboles.



6.1 Découpage en blocs

Le calcul de la DCT ne peut pas se faire sur une image entière d'une part parce que cela générerait trop de calculs et d'autre part parce que le signal de l'image doit absolument être représenté par une matrice carrée. Dès lors, le groupe JPEG impose la décomposition de l'image en blocs de 8 pixels sur 8 pixels. La méthode de compression sera donc appliquée indépendamment sur chacun des blocs.

6.2 Transformée DCT

La DCT est une transformée fort semblable à la FFT. Elle prend un ensemble de points d'un domaine spatial et les transforme en une représentation équivalente dans le domaine fréquentiel. Cette méthode permet de décrire chaque bloc en une carte de fréquences et en amplitudes plutôt qu'en pixels et couleurs. La valeur d'une fréquence reflète l'importance et la rapidité d'un changement, tandis que la valeur d'une amplitude correspond à l'écart associé à chaque changement de couleur.

La transformée DCT, ainsi que la transformée inverse, s'expriment mathématiquement par:

$$F(u,v) = \frac{1}{4} C(u)C(v) \left[\sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

$$f(x,y) = \frac{1}{4} \left[\sum_{u=0}^{7} \sum_{v=0}^{7} C(u)C(v)F(u,v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2x+1)v\pi}{16} \right]$$

$$\text{where} \quad C(u), C(v) = \begin{cases} 1/\sqrt{2}, & \text{for } u, v = 0, \\ 1, & \text{otherwise} \end{cases}$$

6.2 La quantification

La quantification est l'étape dans laquelle on perd réellement des informations (et donc de la qualité visuelle), c'est qui fait gagner de la compression (contrairement à la DCT, qui ne compresse pas). La quantification consiste à diviser la matrice retournée par la DCT, par une autre, appelée matrice de quantification, et qui contient 8x8 coefficients. Le but est ici d'atténuer les hautes fréquences, c'est-à-dire celles auxquelles l'œil humain est très peu sensible. Ces fréquences ont des amplitudes faibles, et elles sont encore plus atténuées par la quantification (les coefficients sont même ramenés à 0).

$$F^*(u,v) = \left\lfloor \frac{F(u,v) + \left\lfloor \frac{Q(u,v)}{2} \right\rfloor}{Q(u,v)} \right\rfloor \cong \left(\frac{F(u,v)}{Q(u,v)} \right)$$

$$\hat{F}(u,v) = F^*(u,v) \cdot Q(u,v)$$

Exemple:

Matrice de pixels d'entrée

140	144	147	140	140	155	179	175
144	152	140	147	140	148	167	179
152	155	136	167	163	162	152	172
168	145	156	160	152	155	136	160
162	148	156	148	140	136	147	162
147	167	140	155	155	140	136	162
136	156	123	167	162	144	140	147
148	155	136	155	152	147	147	136

Matrice DCT quantifiée

403	-4	2	-1	2	-1	-1	-1
4	-5	3	-1	-1	1	1	0
-1	-3	0	0	-1	0	-1	0
-1	0	1	-1	0	0	0	0
0	1	1	0	-1	1	1	1
0	0	-1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Matrice DCT

1210	-18	15	-9	23	-9	-14	-19
21	-34	26	-9	-11	11	14	7
-10	-24	-2	6	-18	3	-20	-1
-8	-5	14	-15	-8	-3	-3	8
-3	10	8	1	-11	18	18	15
4	-2	-18	8	8	-4	1	-7
9	1	-3	4	-1	-7	-1	-2
0	-8	-2	2	1	4	-6	0

Matrice DCT déquantifiée (décompression)

1209	-20	14	-9	22	-13	-15	-17
20	-35	27	-11	-13	15	17	0
-7	-27	0	0	-15	0	-19	0
-9	0	13	-15	0	0	0	0
0	13	15	0	-19	21	23	25
0	0	-17	0	0	0	0	0
15	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Matrice de quantification

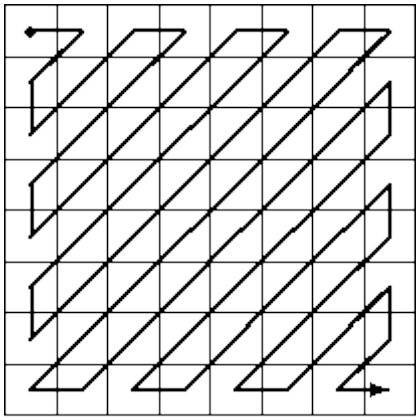
3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Matrice de pixels de sortie (décompression)

142	143	154	141	133	153	179	179
139	152	129	151	144	154	163	181
150	156	139	166	162	163	154	172
163	145	160	153	151	153	145	154
168	150	156	145	140	139	141	159
148	164	133	164	158	140	136	163
130	159	123	164	165	140	134	145
148	156	140	148	159	146	153	141

6.3 Le codage

La dernière étape de la compression JPEG est le codage de la matrice DCT quantifiée. Ce codage est réalisé sans perte d'informations. Le codage du reste de la matrice DCT quantifiée va se faire en parcourant les éléments dans l'ordre imposé par une séquence particulière appelée séquence zigzag.



Cette manière de balayage a la propriété de parcourir les éléments en commençant par les basses fréquences et de traiter les fréquences de plus en plus hautes. Puisque la matrice DCT

quantifiée contient beaucoup de composantes de hautes fréquences nulles, l'ordre de la séquence zigzag va engendrer de longues suites de 0 consécutifs. Deux mécanismes sont mis en œuvre pour comprimer la matrice DCT quantifiée. D'une part, les suites de valeurs nulles sont simplement codées en donnant le nombre de 0 successifs. D'autre part, les valeurs non nulles seront codées en utilisant une méthode statistique de type Huffman ou arithmétique.

6.4 La décompression

Les étapes de la décompression s’effectuent dans l’ordre inverse de la compression suivant les méthodes définies précédemment.