

## **Résumé :**

Ce polycopié propose une introduction structurée au traitement numérique des signaux visuels et sonores, en lien avec les applications contemporaines de l'analyse de données multimédias. Il aborde les fondements théoriques et les outils pratiques nécessaires pour représenter, transformer, analyser et compresser des signaux tels que les images, l'audio et la vidéo.

Le cours débute par la représentation numérique des données et les modèles de couleur, éléments essentiels pour comprendre les formats d'images et leur traitement. Il se poursuit avec les bases de l'analyse fréquentielle, notamment la transformée de Fourier et la convolution, qui permettent de caractériser le contenu spectral des signaux.

Une attention particulière est portée au filtrage spatial et fréquentiel des images, en lien avec les techniques classiques d'amélioration ou de détection de contours. Le principe de la compression JPEG est ensuite détaillé, en tant qu'exemple concret combinant transformation, quantification et codage entropique.

Le polycopié traite enfin des spécificités de l'analyse de l'audio et de la vidéo, en mettant en évidence les méthodes temporelles-fréquentielles (comme les spectrogrammes ou le suivi de mouvement) et les contraintes liées à la perception humaine.

Ce document s'adresse à des étudiants en master ou en école d'ingénieurs, souhaitant acquérir des compétences fondamentales pour explorer, interpréter et manipuler des données multimédias dans le contexte de l'intelligence artificielle, de la vision par ordinateur ou de l'ingénierie du signal.

## **Mots clés :**

Traitement du signal, Analyse fréquentielle, Transformée de Fourier, Convolution, Filtrage spatial, Filtrage fréquentiel, Représentation numérique, Modèles de couleur, Compression d'image, JPEG, Transformée en cosinus discrète (DCT), Analyse audio, Analyse vidéo, Vision par ordinateur, Intelligence artificielle

## الملخص

يُقدّم هذا الملخص الموجز مدخلاً منظماً إلى المعالجة الرقمية للإشارات البصرية والسمعية، وذلك في إطار التطبيقات المعاصرة لتحليل البيانات المتعدّدة الوسائط. ويتناول الأسس النظرية والأدوات العملية الضرورية لتمثيل وتحويل وتحليل وضغط الإشارات مثل الصور والصوت والفيديو.

يبدأ المقرّر بموضوع التمثيل الرقمي للبيانات ونماذج الألوان، وهي عناصر أساسية لفهم صيغ الصور ومعالجتها. ويتبع ذلك استعراض لأساسيات التحليل الترددي، لاسيّما تحويلة فورييه والالتفاف، واللّتين تُستخدمان لتوصيف المحتوى الطيفي للإشارات.

يُخصّص جزء مهم لشرح عملية التصفية في المجالين المكاني والترددي للصور، وذلك بالاعتماد على تقنيات كلاسيكية لتحسين، باعتباره مثالاً عملياً يجمع بين التحويل، الصورة أو لاكتشاف الحواف. كما يتم تفصيل مبدأ ضغط الصور وفق معيار والتكميم، والترميز الإحصائي.

ويختتم المقرّر بدراسة خصائص تحليل الإشارات السمعية والمرئية، مع التركيز على الأساليب الزمنية-الترددية (مثل المقياس الطيفي أو تتبّع الحركة) والقيود المرتبطة بالإدراك البشري.

هذا المقرّر موجّه إلى طلبة الماستر أو المدارس الهندسية، الذين يرغبون في اكتساب كفاءات أساسية في استكشاف وتفسير ومعالجة البيانات المتعدّدة الوسائط، في سياق الذكاء الاصطناعي، أو الرؤية الحاسوبية، أو هندسة الإشارات.

## الكلمات المفتاحية

معالجة الإشارة، التحليل الترددي، تحويلة فورييه، الالتفاف، التصفية المكانية، التصفية الترددية، التمثيل الرقمي، نماذج الألوان، تحويلة جيب التمام المنفصل، تحليل الصوت، تحليل الفيديو، الرؤية الحاسوبية، ضغط الصور، الذكاء الاصطناعي

# Sommaire du Polycopié

## Préambule

### Chapitre I : Représentations numériques, couleur

- 1- Traitement d'image et intelligence artificielle
- 2- L'image numérique
  - 2.1 Acquisition
- 3- Composition et caractéristique de l'image numérique
  - 3.1 Le pixel
  - 3.2 La définition
  - 3.3 La résolution
- 4- La couleur
  - 5.1 La Trichromie
    - L'espace RGB
    - L'espace XYZ
    - L'espace HLS (HSV)
- 5- Codage des images
  - 5.1 L'image binaire
  - 5.2 L'image en 256 nuances de gris
  - 5.3 L'image couleur
  - 5.4 L'image couleur indexée
- 6- Représentation des images
  - 6.1 Numérisation des images
  - 6.2 Définition
  - 6.3 L'échantillonnage
  - 6.4 La quantification
  - 6.5 Stockage des images

### Chapitre II : Analyse de Fourier, convolution, filtrage

- 1- L'histogramme
  - 1.1 Définition
  - 1.2 Opérations sur l'histogramme
- 2- La transformée de Fourier
  - 2.1 Représentation spatiale des images 2D
  - 2.2 Représentation fréquentielle d'une image 2D
  - 2.3 Interprétation de la transformée de Fourier discrète
- 3- La convolution
  - 3.1 Topologie de l'image
  - 3.2 Définition et calcul d'une convolution

- 4- Filtrage d'image
- 4.1 Définition d'un filtre
- 4.2 Notion de bruit
- 4.3 Les types de filtres
- 4.4 Filtrage dans le domaine spatial
- 4.4.1 Filtre moyenneur
- 4.4.2 Filtre gaussien

### **Chapitre III : Théorie de l'information et compression d'image (JPEG)**

- 1- Entropie de l'image
- 1.1 Définition et interprétation
- 1.2 Entropie et redondance
- 2- Codage MICD
- 3- Notions de compression
- 3.1 Codage par transformation
- 3.2 Transformée en cosinus discrète (DCT)
- 3.3 Quantification
- 3.4 Codage entropique
- 4- JPEG – Schéma de compression
- 4.1 Séparation DC/AC
- 4.2 Scannage zig-zag
- 4.3 Tables de quantification
- 5- JPEG2000 (aperçu)

### **Chapitre IV : Analyse et compression audio**

- 1- Codage perceptuel et sous-bandes
- 1.1 Modèle psycho-acoustique
- 1.2 Découpage en bandes (filtrage QMF)
- 2- Normes MPEG audio
- 2.1 Couche 1 – principe et structure
- 2.2 Couche 2 – amélioration spectrale
- 2.3 Couche 3 – codage à transformée (MDCT)
- 3- Application de la quantification et du codage de Huffman

### **Chapitre V : Analyse et compression vidéo**

- 1- Structure temporelle de la vidéo
- 1.1 Notion de trame (I, P, B)
- 1.2 Prédiction temporelle et compensation de mouvement
- 2- Normes de compression vidéo
- 2.1 MPEG-1, MPEG-2

- 2.2 Motion JPEG
- 2.3 H.264
- 3- Conteneurs multimédias (AVI, MP4, MKV, etc.)

**Pniversité Echahid Hamma Lakhder El Oued.**

**Intitulé du Master : Artificial Intelligence & Data Science**

**Intitulé de la matière : Extraction d'Information dans les documents textes, audio, Vidéo**

## **Préambule**

Un environnement multimédia est un ensemble de matériels et de logiciels permettant :

- De créer
- De stocker et d'organiser
- De consulter et modifier en temps réel
- De transmettre

Des documents structurés de façon homogène comportant des informations textuelles, de la voix, des images fixes ou animées, codées numériquement.

L'information qui transite sur les réseaux de télécommunication consiste en messages de types divers : textes, sons, images fixes ou animées, vidéo, etc. La forme que revêt cette information est commode pour une communication directe et classique (conversation, échange sur papier, ...) lorsque les interlocuteurs sont en présence. Quand ils sont distants l'un de l'autre, l'emploi des réseaux de télécommunication est une manière moderne de résoudre la transmission d'informations. Toutefois, pour les nécessités du transport, la transmission d'un message nécessite un encodage en signaux de type électrique ou électromagnétique : l'émetteur et le récepteur sont, de nos jours, des ordinateurs. La voie de transmission peut être une simple liaison directe entre émetteur et récepteur ou beaucoup plus complexe dans le cadre d'un ou plusieurs réseaux de télécommunications. Les signaux sont les véhicules de transport de l'information.

Nous distinguons diverses applications du multimédia dont nous pouvons citer :

- Compression d'images
- Amélioration visuel (Télévision)
- Imagerie Médicale
- Identification empreinte digitale & Reconnaissance de visage
- Authentification
- Aide à la conduite
- Imagerie satellitaire
- Indexation
- Production audiovisuelle numérique (CD, film, vidéo)
- Edition Hors ligne (CD/DVD)
- Edition En ligne (site web, télévision interactive)
- Les dispositifs techniques du spectacle vivant (danse, concert, théâtre, installations interactives)

## **Chapitre I : Représentations numériques, couleur**

L'information existe sous des formes diverses [Gonzalez & Woods, 2018]. Pour la manipuler et, en particulier, la transporter, nous sommes amenés à la coder. De nos jours, l'information est souvent

présentée dans des documents composites, comme une page Web, où simultanément peuvent être présentés : un texte, une image fixe, un clip vidéo. L'information est, en effet, présentée sous forme multimédia. Chaque type d'information possède son système de codage, mais le résultat est le même : une suite de 0 et de 1. Le transport de l'information consiste alors à transmettre des bits, quel que soit la signification du train de bits transmis.

Nous pouvons affirmer que l'information existe sous des formes diverses, citons :

- **Image fixe**

Système : télécopie (appareil électronique convertissant l'image de documents en impulsions électriques pour les transmettre à un destinataire au travers d'une ligne téléphonique. À la réception, nous utilisons un appareil similaire à celui de l'émission pour faire la conversion inverse et réaliser l'impression d'un document identique à l'original)

Codeur : scanner

Décodeur : interpréteur de fichier

Transmission : signaux analogiques (issues de la nature) et numériques

- **Parole**

Système : téléphone

Codeur : microphone

Décodeur : écouteur

Transmission : signaux analogiques et Numériques

- **Télévision**

Système : télévision hertzienne (correspondant à une diffusion par l'intermédiaire d'ondes électromagnétiques, en effet, les signaux de télévision ou Radio (FM) transitent dans l'espace, sans support matériel, puis seront reçus par une antenne appropriée)

Codeur : caméra

Décodeur : récepteur TV + antenne

Transmission : signaux analogiques

- **Données informatiques**

Système : réseaux de téléinformatique (pour effectuer la transmission des informations d'un ordinateur à un autre ordinateur via le réseau téléphonique, le signal sort tout d'abord de la carte réseau de l'ordinateur, pénètre dans un modem, puis du modem vers le réseau téléphonique, puis du réseau téléphonique vers un autre modem lié à la carte réseau de l'autre ordinateur).

Codeur : contrôleur de communication + ETCD (équipement terminal de circuit de données qui permet d'adapter le flux des données aux conditions de la ligne ainsi que de réaliser la transformation analogique / numérique ou bien numérique / analogique)

Décodeur : contrôleur de communication + ETCD (pouvant s'agir d'un modem).

Transmission : signaux analogiques ou numériques

Suite à un transport et un stockage en numérique tout signal (vidéo ou audio) devra revenir à sa forme analogique de départ. Exemple : un signal audio sera converti du numérique → à l'analogique pour

ensuite être amplifié car les oreilles humaines ne savent entendre en numérique. A cet effet, le numérique ne sert (dans le cas d'un signal audio ou vidéo) qu'au transport et au stockage de données.

## 1- Traitement d'image et intelligence artificielle

Dans le domaine de la vision artificielle, nous distinguons le traitement d'images juste après les étapes d'acquisition et de numérisation dans le but d'assurer les transformations d'images ainsi que la phase de calcul qui permet la réalisation d'une interprétation des images traitées. En effet, l'appel à l'intelligence artificielle est nécessaire pour la manipulation des connaissances sur les informations dont nous disposons à propos de ce que représentent les images traitées.

Dans la conception moderne de l'Intelligence Artificielle, la machine agit sur le monde extérieur, éventuellement se déplace, et aussi perçoit son environnement pour pouvoir s'y adapter. Réciproquement, le traitement d'image et la vision tirent parti des connaissances et des techniques d'intelligence artificielle pour gérer l'adaptation à un environnement changeant, l'information incertaine, les systèmes hétérogènes de connaissances et les différents niveaux de prise de décision.

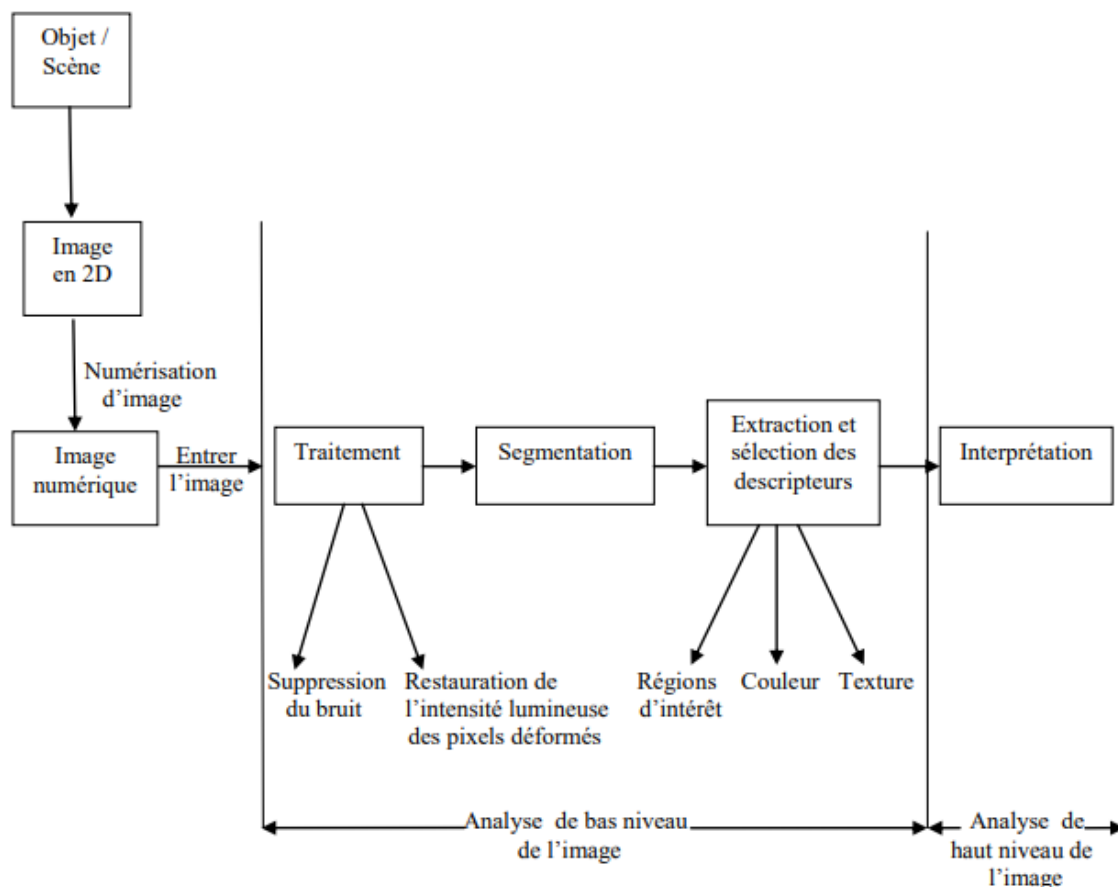


Figure 1.1 Les niveaux d'analyse d'images.

## 2- L'image numérique

### 2.1 Acquisition

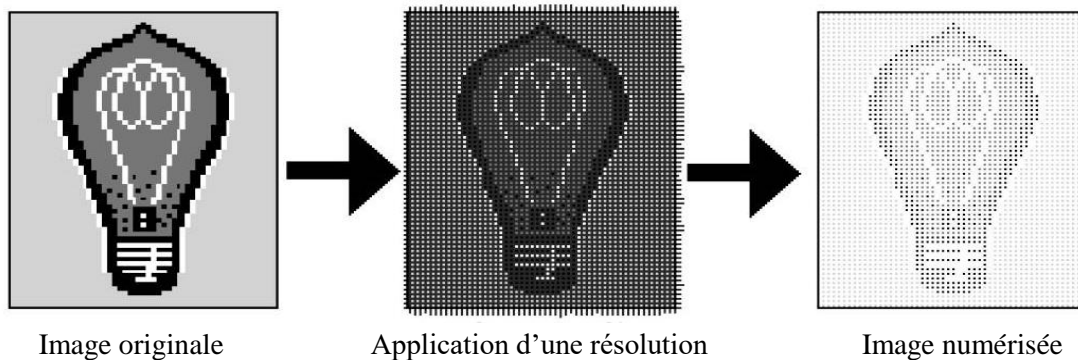
Une image est une représentation planaire d'une scène ou d'un objet situé dans un espace à 3 dimensions (+ la dimension temporelle). L'acquisition, le traitement et la restitution d'une image

s'apparentent à une chaîne de traitement (traitement d'un signal) avec toutes ses problématiques notamment les capteurs : capteurs (numériseurs).

Une image numérique est constituée d'une grille rectangulaire d'échantillonnage dont les constituants sont des pixels portant des informations sur l'intensité lumineuse des différents lieux au sein de l'image.

Lors de l'acquisition d'une image, une lumière est prise ensuite chaque pixel du capteur prend une quantité de lumière (couleur) et ces informations seront enregistrées sur la carte mémoire (numérisation) grâce au processeur de l'appareil photo.

De ce fait, une image numérique est représentée par une matrice bidimensionnelle, dont les éléments sont des nombres naturels correspondant à des niveaux de quantification dans l'échelle de l'intensité lumineuse [Szeliski, 2022]. Tout d'abord une image d'entrée est capturée par une caméra de télévision en (2D) et numérisée, étant décrite par une fonction d'image  $f(x, y)$  dont la valeur est l'intensité lumineuse en fonction de deux paramètres  $x, y$ , représentant les coordonnées de l'emplacement dans l'image.



Une image n'est qu'une représentation imparfaite d'une scène et élaborer une image correspond à la volonté de proposer une entité observable par l'œil humain. Le traitement d'image consiste en à partir des informations initiales, de chercher souvent à extraire des informations plus pertinentes, et à les interpréter. Nous distinguons ainsi des prétraitements, des traitements « bas-niveau » (appliqués aux pixels, sans leur donner de signification) et « haut-niveau » : analyse et interprétation.

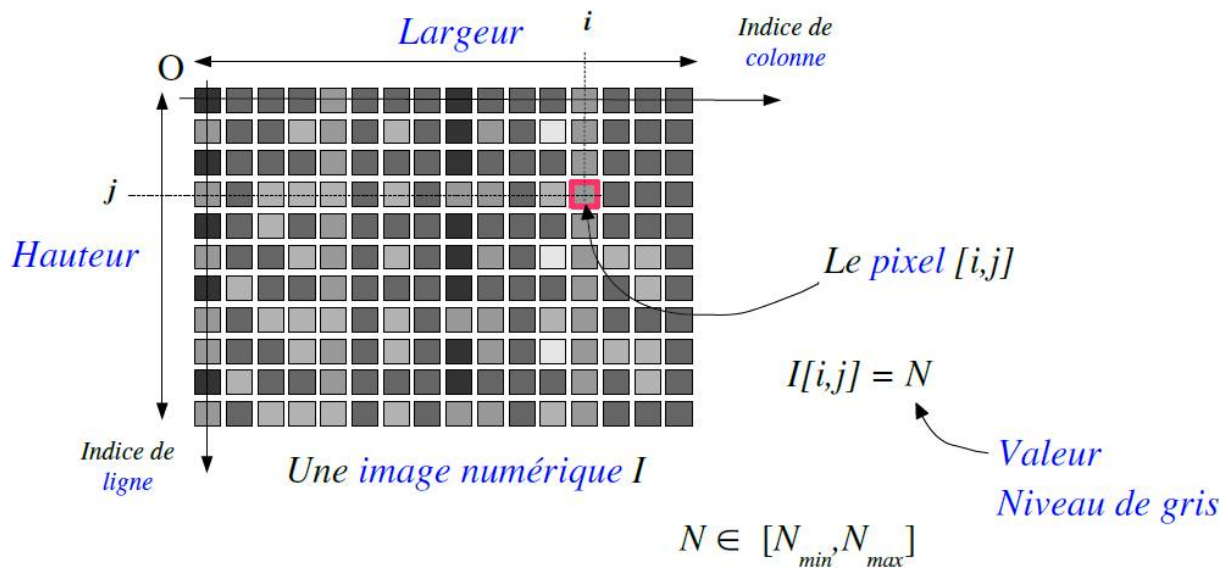
### 3- Composition et caractéristique de l'image numérique

#### 3.1 Le pixel

Une image numérique est constituée d'un ensemble de points appelés pixels (abréviation de PICTURE Element) pour former une image. Le pixel représente ainsi le plus petit élément constitutif d'une image numérique [Gonzalez & Woods, 2018]. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image comme mentionné sur la figure 1.2.

Nous pouvons considérer qu'un pixel n'a pas de dimension car si nous regardons une photo, un pixel d'un objet situé près de la caméra correspond à une taille plus petite qu'un pixel d'un objet situé loin de la caméra, même si ces deux pixels font partie de la même image.

Aussi, le pixel n'a pas de résolution ou plus juste une taille et devrait s'adapter avec la surface sur laquelle il est projeté. Donc, les pixels s'étirent pour remplir l'écran ou se combinent pour y figurer entièrement lorsqu'on aura plus de pixels que l'écran peut représenter.



$$(N_{max} - N_{min}) = \text{nombre de niveaux de gris}$$

**Figure 1.2** Une image numérique

### 3.2 La définition

Nous appelons définition le nombre de points (pixels) constituant une image : c'est le nombre de colonnes de l'image que multiplie son nombre de lignes. Une image possédant 10 colonnes et 11 lignes aura une définition de 10 x 11.

- **Les Principaux formats d'affichage :**

L'image s'affiche sur un écran (appelé aussi moniteur), il s'agit d'un périphérique de sortie permettant de fournir une représentation visuelle. Ces informations proviennent de l'ordinateur, mais de façon " indirecte ". En effet le processeur n'envoie pas directement les informations au moniteur, mais traite les informations provenant de sa mémoire vive (RAM), puis les envoie à une carte graphique qui est chargée de convertir les informations en impulsions électriques qu'elle envoie au moniteur.

Pour afficher ces images, des formats d'affichages standards ont été définis en fonction de l'évolution des capacités matérielles des cartes graphiques et des écrans. Voici les principaux :

- CGA (320 x 200) 4 couleurs.
- VGA (640 x 480) 16 couleurs.
- SVGA (800x600) 256 couleurs.
- XGA (1024 x 768) 256 couleurs.
- SXGA (1280 x 1024) en 16 millions de couleurs.

- **Calcul du nombre total des pixels dans une image :**

Nombre total des pixels = colonnes x lignes.

Ex : 10x11= 110 pixels au total pour l'image ci-dessus.

### 3.3 La résolution

C'est le nombre de pixels que l'on peut loger sur une longueur donnée (en pouce). Elle est exprimée en points par pouce (PPP, en anglais : DPI pour Dots per Inch).

Exemple 1 : une image de résolution 20 pixels par centimètre : la taille de chaque pixel est telle que sur une ligne de 1 cm nous pouvons en loger 20 (la taille des pixels étant le plus souvent carrée, nous logons le même nombre de pixels sur les deux axes).

Exemple 2 : une image de résolution 40 pixels par centimètre : il y a donc deux fois plus de pixels pour chaque centimètre linéaire. A cet effet, chaque pixel est deux fois plus petit.

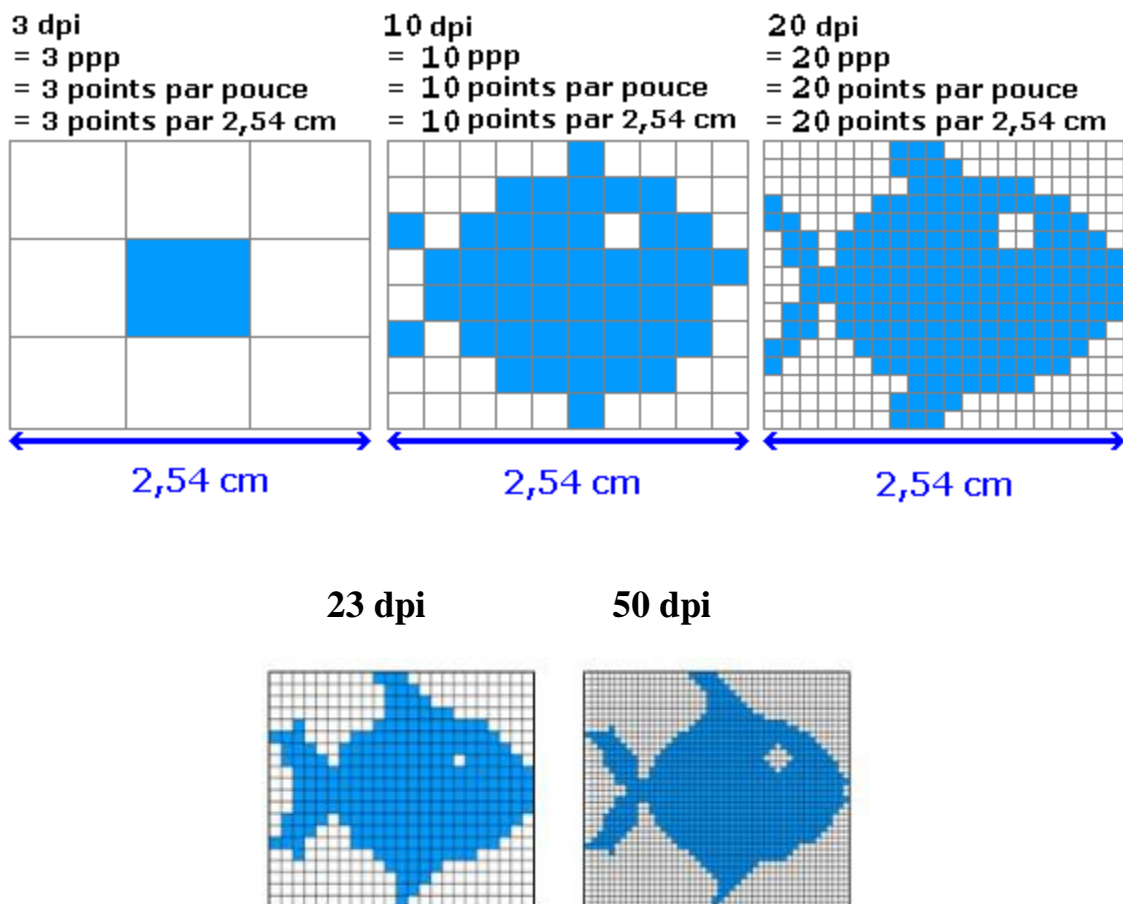
D'après les deux exemples ci-dessus : il est aisé de comprendre que plus le pixel est petit, meilleure sera la qualité de l'image. En effet, plus la densité des pixels est grande (beaucoup de pixels logés sur une longueur) meilleure sera la qualité de l'image (élimination de la pixelisation).

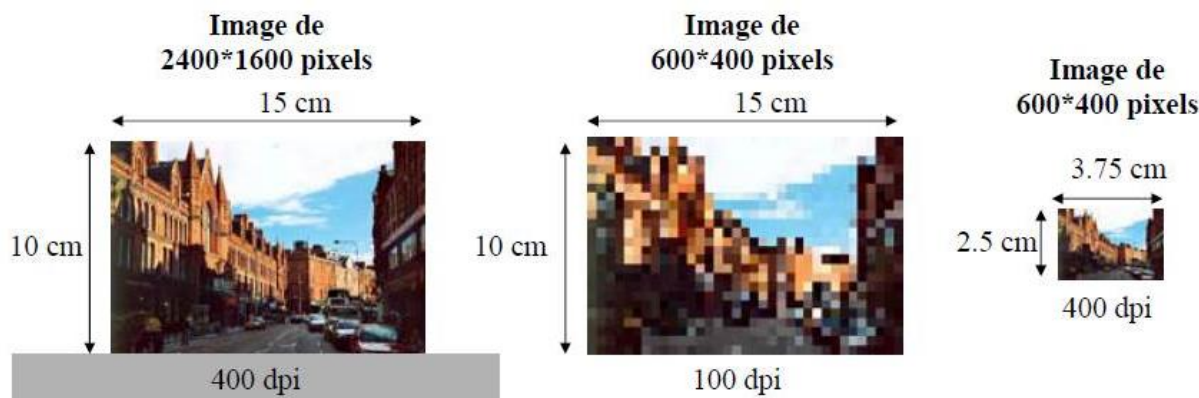
Un pouce mesure 2.54 cm, c'est une unité de mesure britannique. La résolution permet ainsi d'établir le rapport entre la définition en pixels d'une image et la dimension réelle de sa représentation sur un support comme mentionné sur la figure 1.3 [Burger & Burge, 2016].

Plus il y aura de pixels par pouce, plus il y aura de l'information dans l'image car la résolution d'une image définit la netteté et la qualité de l'image.

Exemple 3 : une image à 300 PPP est seize fois plus lourde que la même image à 75 PPP.

Il est utile de mentionner qu'un pixel n'a pas de taille prédéfinie, donc si nous affichons une image à l'écran, chaque pixel constitutif de l'image occupera un pixel de l'écran.





**Figure 1.3 :** Diverses résolutions d'image

La résolution est souvent confondue avec la " définition ", détermine par contre le nombre de points par unité de surface ; un pouce représentant 2.54 cm. La résolution permet ainsi d'établir le rapport entre le nombre de pixels d'une image et la taille réelle de sa représentation sur un support physique. Une résolution de 300 dpi signifie donc 300 colonnes et 300 rangées de pixels sur un pouce carré ce qui donne donc 90000 pixels sur un pouce carré.

- **Quelques exemples de résolutions fréquemment utilisées**

Ecrans d'ordinateur : 72 dpi. C'est aussi dans cette résolution que sont les images sur Internet. Résolution non adaptée à l'impression !

- fax : en générale en 200 dpi.

- Imprimantes grand public : entre 360 dpi et 1400 dpi. Cela permet d'obtenir une qualité tout à fait honorable pour tous les travaux courants (courriers, rapports, etc....)

- Scanners grand public : 300, 600 ou 1200 dpi.

- Matériel d'impression professionnel : aux minimum 4800 dpi (impression de qualité et grandes tailles pour les affiches).

#### 4- La couleur

Les couleurs trouvent leur origine dans la séparation de la lumière blanche naturelle en composantes absorbées et composantes réfléchies. Toute source lumineuse visible est composée d'un mélange d'ondes électromagnétiques cohérentes (i.e. couleurs pures), dont la longueur d'onde est comprise entre 0,4  $\mu\text{m}$  (violet) et 0,7  $\mu\text{m}$  (rouge) :



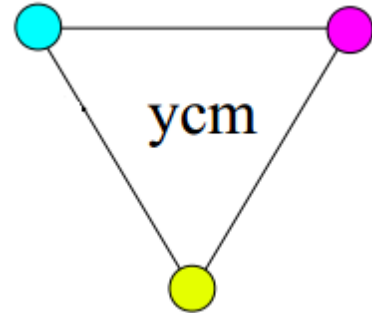
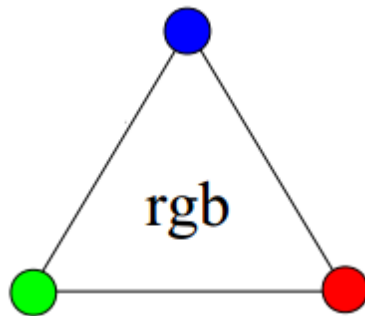
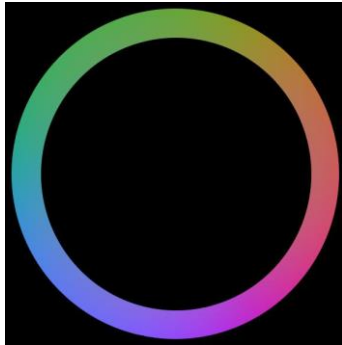
Spectre de la lumière blanche

La question principale que nous allons considérer dans la suite est : « Comment représenter ces couleurs dans un espace qui soit à la fois facile à manipuler et qui soit pertinent en termes d'analyse d'images couleur ? »

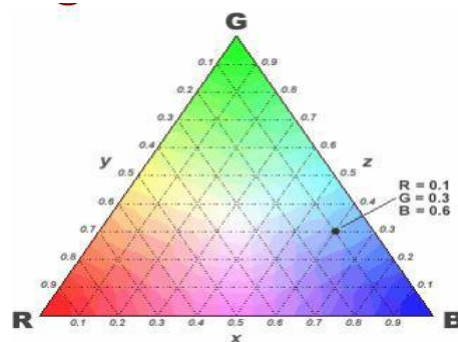
## 5.1 La Trichromie

Le principe de la trichromie réside dans la découverte qu'il suffit d'un triplet de couleurs pures pour restituer, par mélange de ces 3 composantes que nous pouvons qualifier de primaires, l'intégralité des couleurs.

Le cercle de Newton peut ainsi se « résumer » en un triangle de primaires.



En remplissant le triangle de primaires selon la loi des aires, nous obtenons le triangle de Maxwell :

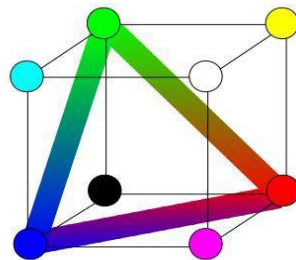


La trichromie permet donc de voir l'ensemble des couleurs comme un espace vectoriel de dimension 3, où 3 sources lumineuses primaires (r,g,b) varient en intensité entre 0 et 100%. Ce principe de synthèse additive de la couleur se retrouve dans la plupart des dispositifs lumineux de restitution de la couleur : CRT, LCD, Plasma.

- **L'espace RGB**

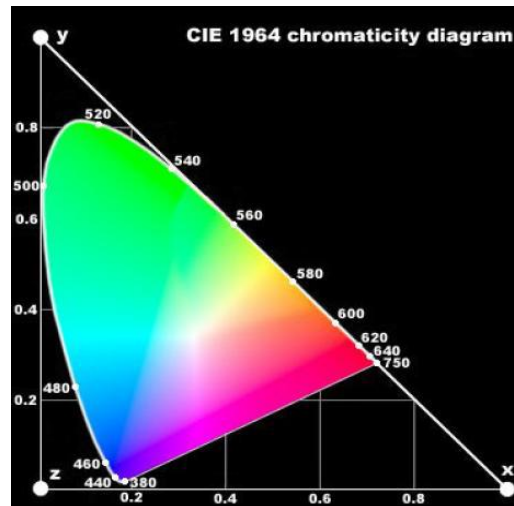
L'espace RGB est l'espace vectoriel engendré par les 3 composantes primaires (Rouge, Vert, Bleu) [Gonzalez & Woods, 2018].

L'ensemble des couleurs produites se représente comme l'intérieur d'un cube tel qu'il est mentionné sur le triangle chromatique suivant :



- **L'espace XYZ**

L'espace XYZ (CIE 1931) est défini à partir d'une transformation linéaire sur l'espace RGB telle que toutes les couleurs du spectre visible soient contenues dans le triangle xyz. Les coordonnées XYZ des couleurs naturelles ont ainsi toujours des valeurs positives.



- **L'espace HLS (HSV)**

La perception humaine de la couleur est une réaction subjective qui peut être caractérisée en termes de luminosité, de teinte et de saturation [O'Shaughnessy, 2000].

- La luminosité correspond à une sensation traduite par des vocables comme clair, foncé, lumineux, sombre...

- La teinte ou tonalité chromatique correspond aux dénominations telles que rouge, vert, bleu, ... Elle correspond à la longueur d'onde dominante d'une couleur. Le blanc, le noir ou les gris sont dites neutres ou achromatiques.

- La saturation est une grandeur estimant le niveau de coloration d'une teinte indépendamment de la luminosité. Elle représente la pureté de la couleur perçue comme vive, pâle.

Le passage de RGB à HSV se fait par une transformation non linéaire. Plusieurs opérateurs ont été proposés pour la conversion [Smith, 2007].

## **5- Codage des images**

Un pixel est codé suivant la qualité de l'image :

### **5.1 L'image binaire**

Dans une image en noir et blanc (image binaire), un seul bit suffit pour coder le point (0 pour noir, 1 pour blanc) comme mentionné sur la figure suivante :

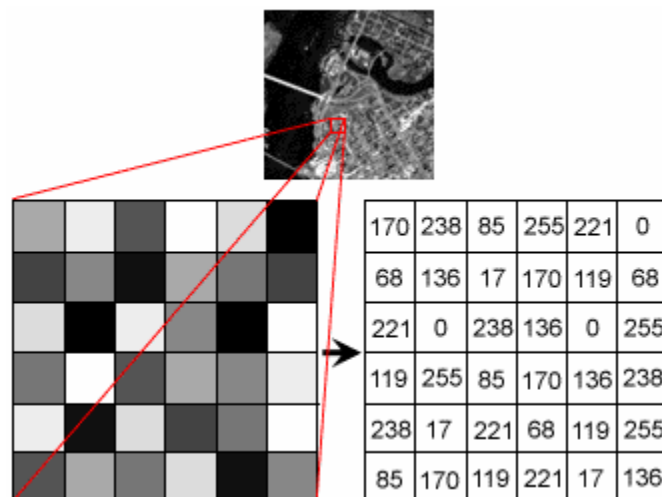
1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	0	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1
1	0	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1
1	0	1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1

**Figure 1.4** Image binaire

## 5.2 L'image en 256 nuances de gris

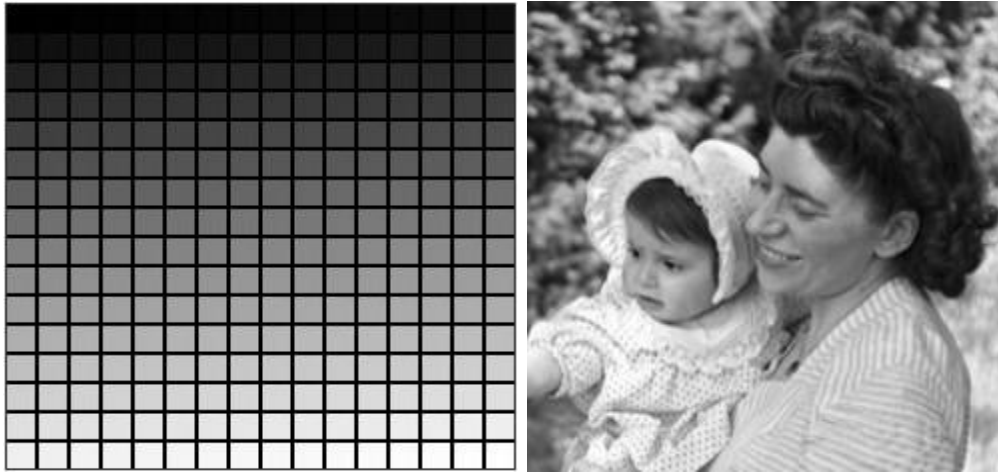
Dans une image en 256 nuances de gris, chaque pixel est représenté par un octet (8 bits). Nous ne codons ici que le niveau de l'intensité lumineuse, généralement sur un octet ( $2^8 = 256$  valeurs). Par convention, la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale) :

Nous présentons sur la figure 1.5, les niveaux de gris relatifs aux pixels d'une partie d'une image.



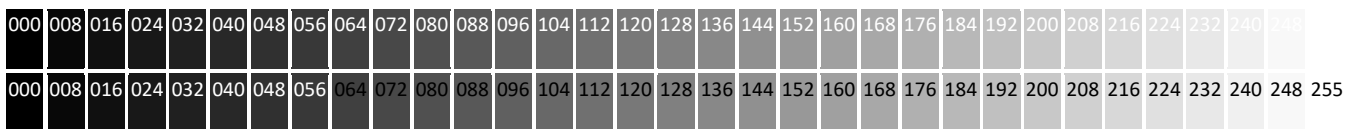
**Figure 1.5** Niveaux de gris relatifs aux pixels d'une partie d'une image

Quant à la figure ci-dessous, nous présentons un exemple de photo en 8 bpp.



**Figure 1.6** Nuances de 256 gris (à gauche), exemple de photo possible en 8 bpp (à droite)

Il est utile d'avoir une idée sur les 256 nuances de gris comme mentionné sur la figure 1.7.



**Figure 1.7** Nuances de 256 gris

Ce procédé est fréquemment utilisé pour reproduire des photos en noir et blanc ou du texte.

### 5.3 L'image couleur

Dans une image couleur, nous montrons que la couleur peut être exprimée comme une combinaison linéaire de trois couleurs de base, par exemple Rouge(R), Vert(V), Bleu(B). Ainsi une couleur quelconque  $x$  est exprimée comme  $x = aR + bV + cB$ . Où  $a$ ,  $b$ ,  $c$  sont des doses de couleurs de base. Chacun de ces éléments dispose de nuances allant de 0 à 255 : 256 couleurs. Pour avoir 256 couleurs, il faut 8 bits, donc 1 octet. Comme il y a 3 éléments différents RVB, il nous faut donc 3 octets pour rendre bien compte de toutes les nuances.

Le codage RVB (ou RGB en anglais) est celui utilisé dans de nombreux périphériques numériques (Scanners, appareils photos, écrans...) [Gonzalez & Woods, 2018]. Le principe est simple, un octet sert à coder un nombre compris entre 0 et 255 (d'où 256 valeurs) qui correspond à la valeur de la composante rouge, verte ou bleue du pixel (0 étant l'absence de cette composante, 255 la saturation). Le mélange de ces trois composantes donne la couleur finale.

Les 24 bits d'une couleur se décomposent en 3 fois 8 bits, c'est-à-dire 3 octets :

- 8 bits sont consacrés à la teinte primaire rouge ;
- 8 bits sont consacrés à la teinte primaire vert ;
- 8 bits sont consacrés à la teinte primaire bleue.

Usuellement, une bonne image correspond à des doses allant de 0 à 255. Par suite une image couleur de ce type peut être représentée par 3 matrices (une par couleur de base) dont chacune d'elle possède des éléments sur 8 bits, ce qui au total fait 24 bits par pixel. Nous nous rendons vite compte du volume atteint pour des images importantes et de bonne définition. Une image 640x480 en couleur (24 bits)

occupe un volume de 921 600 octets. Nous sommes alors amenés à utiliser des techniques de compression pour réduire la taille des fichiers d'images.

L'une des premières normes d'affichage, la norme CGA, permettait d'afficher des pixels dans une palette de 4 couleurs différentes. Pour cela, il fallait que chaque pixel soit codé par 2 bits.

Par exemple :

Numéro (décimal)	Numéro (binaire)	Couleur
0	00	Noir
1	01	Vert
2	10	Rouge
3	11	Jaune

Ensuite, le nombre de couleurs que nous pouvions représenter à la fois a continué à augmenter. Nous pouvons à présent avoir des images de 16 couleurs. Et puisque  $16 = 2^4$ , il suffit pour cela de coder chaque pixel par une suite de 4 bits. Voici un exemple de palette courante :

N° décimal	N° binaire	Couleur	N° décimal	N° binaire	Couleur
0	0000	Noir	8	1000	Gris foncé
1	0001	Bleu	9	1001	Bleu clair
2	0010	Vert	10	1010	Vert clair
3	0011	Cyan	11	1011	Cyan clair
4	0100	Rouge	12	1100	Rouge clair
5	0101	Magenta	13	1101	Magenta clair
6	0110	Marron	14	1110	Jaune
7	0111	Gris	15	1111	Blanc

Nous pouvons continuer ainsi pour découvrir des images de plus en plus nette grâce à leur grand nombre de couleurs : avec 8 bits par pixels, nous obtenons 256 couleurs (car  $2^8 = 256$ ) ; avec 16 bits, 65536 couleurs ; avec 24 bits, plus de 16 millions de couleurs [Gonzalez & Woods, 2018]; avec 32 bits, plus de 4 milliards. À partir de 24 bits, nous parlons d'ailleurs d'affichage en vraies couleurs tant le réalisme est parfait. A cet effet :

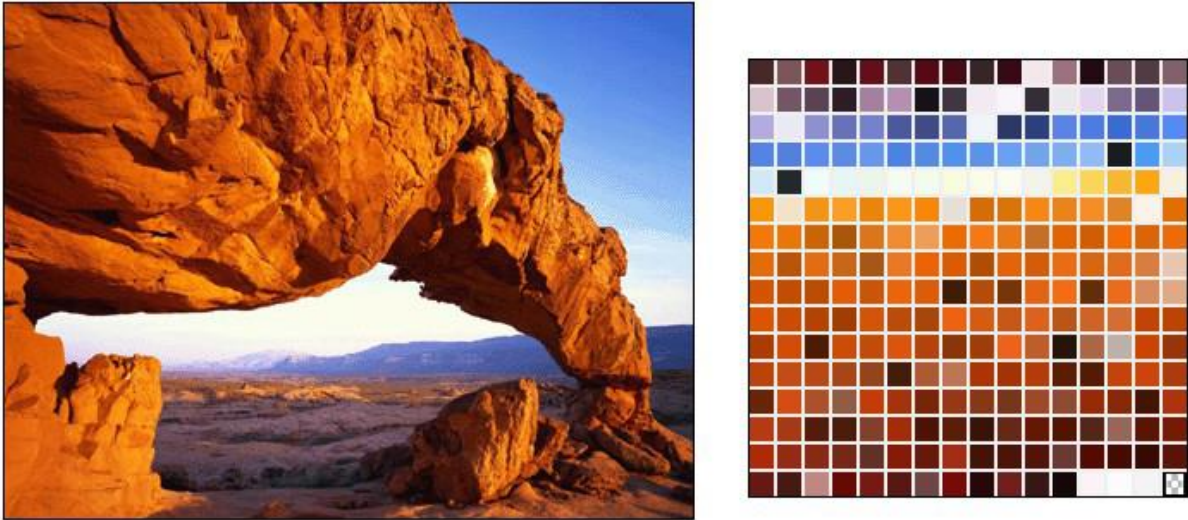
- 1 bit :  $2^1$  possibilités : image noire et blanc
- 2 bits :  $2^2$  possibilités : 4 couleurs
- 4 bits :  $2^4$  possibilités : 16 couleurs
- 8 bits :  $2^8$  possibilités : 256 couleurs
- 24 bits :  $2^{24}$  : 16 777 216 millions de couleurs

#### 5.4 L'image couleur indexée

Permet d'obtenir jusqu'à 256 couleurs fixes, définies à l'avance dans une palette. Il n'utilise qu'une seule couche.

- Codage en 8 bits par pixel (bpp) :  $2^8 = 256$  possibilités
- Chaque pixel peut avoir jusqu'à 256 couleurs fixes possibles

Comme nous pouvons le constater sur la figure 1.8, avec 256 couleurs, certains dégradés de cette image apparaîtront tramés, la qualité est proche de la photo mais il est possible de faire beaucoup mieux.



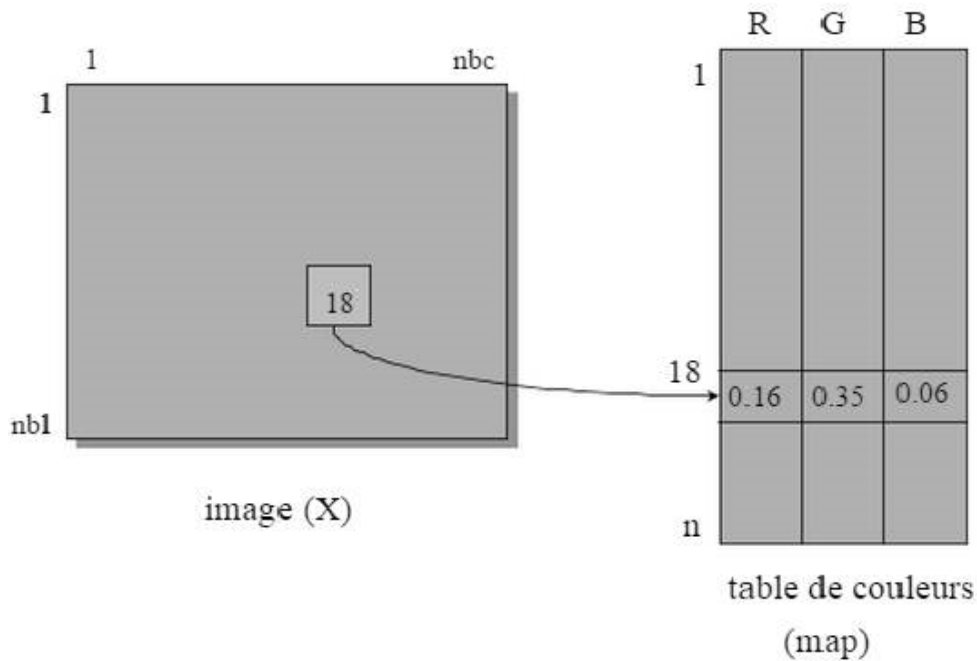
**Figure 1.8** Exemple d'une image couleur indexée

Nous pouvons constater que la manière la plus naturelle de représenter une image couleur consiste à utiliser trois matrices. Toutefois, pour de grandes images, ceci conduit à l'occupation d'un espace mémoire important. A cet effet, Matlab a proposé la représentation indexée vu qu'elle est plus économique pour la représentation des images couleur. En contrepartie, la représentation indexée ne permet de représenter qu'un nombre limité de couleurs.

Nous allons démontrer plus précisément le principe de cette représentation comme indiqué sur la figure 1.8, en effet, Ces couleurs sont mémorisées dans une table de couleurs (colormap) qui est une matrice  $n \times 3$  (où  $n$  est le nombre de couleurs). L'image est alors une matrice contenant des nombres entiers compris entre 1 et  $n$ , chaque entier jouant le rôle d'un index relatif à la table de couleurs.

Exemple :

L'élément représenté sur la figure 1.9 contient l'entier 18, ce qui ramène ou pointe vers la ligne 18 de la table de couleurs pour connaître la couleur du pixel correspondant.



**Figure 1.9** Table de couleurs (colormap)

## 6- Représentation des images

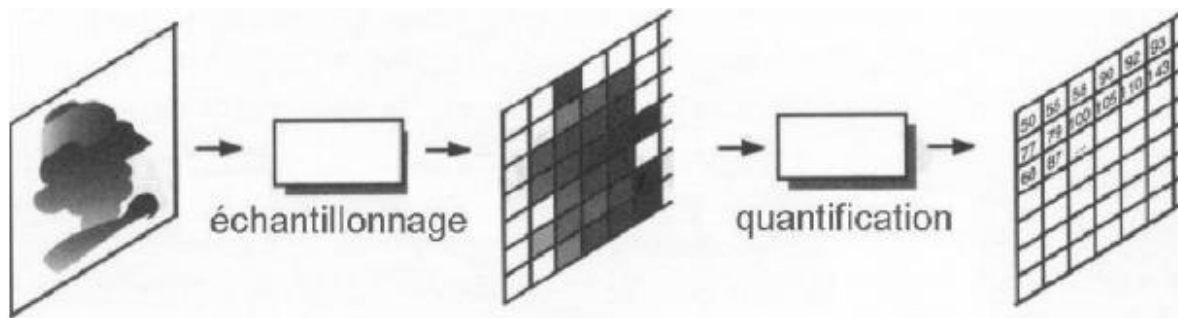
### 6.1 Numérisation des images

La numérisation d'une image est obtenue par l'intermédiaire d'un capteur et d'un numériseur, qui transforment un signal optique en un signal numérique [Szeliski, 2022]. Le signal optique peut être obtenu directement ou par l'intermédiaire d'un système d'agrandissement optique : lentille simple, loupe ou microscope photonique. Le capteur est constitué par un ensemble de capteurs élémentaires, une barrette de CCD (Charge Couple Device), ça peut être une caméra, noir et blanc ou couleur, un appareil photo numérique ou un scanner. Le signal électrique est repris par un convertisseur analogique-digital (carte de numérisation intégrée ou non au capteur) qui transforme les données continues en données numériques codées sur 1, 8, 16 ou 24 bits. Le codage utilisé définit le type d'images (noir et blanc, niveaux de gris ou couleur) et sa " profondeur ".

L'échantillonnage et la quantification sont deux opérations permettant de numériser une image [Oppenheim & Schaffer, 2010] :

- L'échantillonnage, visant à prélever régulièrement des échantillons sur le signal. Il est limité par la capacité du capteur, donc le nombre de pixels disponible.
- La quantification, visant à approcher le signal continu par un ensemble fini de valeurs. Elle est limitée par la quantité de tons (de gris) définie dans l'intervalle.

Nous présentons sur la figure I.10, une schématisation de l'échantillonnage ainsi que de la quantification.



**Figure I.10** Schématisation des processus de l'échantillonnage et de la quantification

Les buts de la numérisation des images consistent en : les visualiser sur un moniteur, les imprimer, les traiter sur un ordinateur ; les stocker sur des supports informatiques ou les transmettre sur des réseaux informatiques.

De l'analogique au numérique :

Image **analogique** : image formée sur la rétine de l'œil.

Image **numérique** : signal numérique composé d'unités élémentaires (pixels) représentant chacun une portion de l'image. Contrairement aux images obtenues à l'aide d'un appareil photo (analogique), ou dessinées sur du papier, les images manipulées par un ordinateur sont numériques (représentées par une série de bits).

## 6.2 Définition

L'image numérique :  $f: [1, M] \times [1, N] \rightarrow I$ , où  $MN$  est le nombre de pixels et  $I$  est un ensemble discret de valeurs.

- Cas où  $I = \{0, 1\} \rightarrow$  c'est une image binaire.
- Cas où  $I = \{0, \dots, P - 1\}$  (souvent  $P = 2^8 = 256$  : codage sur 8 bits)  $\rightarrow$  c'est une image en niveaux de gris.
- Cas où  $I = \{0, \dots, P - 1\} \times \{0, \dots, P - 1\} \times \{0, \dots, P - 1\} \rightarrow$  c'est une image couleur.

Rouge

Vert

Bleu

## 6.3 L'échantillonnage

Comment passer d'un signal continu à un signal discret pouvant être enregistré, analysé et/ou traité ?

Rappel :

- Image continue 2D (ou perçue comme telle) :

$$D \in \mathbb{R} \times \mathbb{R} \rightarrow C$$

Où  $C$  est un espace de couleurs.

- Image discrète 2D :

$$D \in \mathbb{N} \times \mathbb{N} \rightarrow C$$

Où  $C$  est un ensemble fini de couleurs

- Résolution verticale : nombre de lignes dans l'image.
- Résolution horizontale : nombre de colonne dans l'image.
- Résolution spatiale : résolution verticale x résolution horizontale.

- Densité de résolution : nombre de pixels par unité de longueur. S'exprime en ppi (pixels per inch) ou dpi (dots per inch).

La réponse technologique pour les images consiste à utiliser un appareil numérique (ce sont alors les capteurs CCD qui numérisent le signal) ou un scanner pour numériser des photos. La réponse théorique à cette question est la théorie de l'échantillonnage. En effet, l'échantillonnage est une étape fondamentale qui doit tenir compte du contenu informationnel pertinent de l'image à analyser.

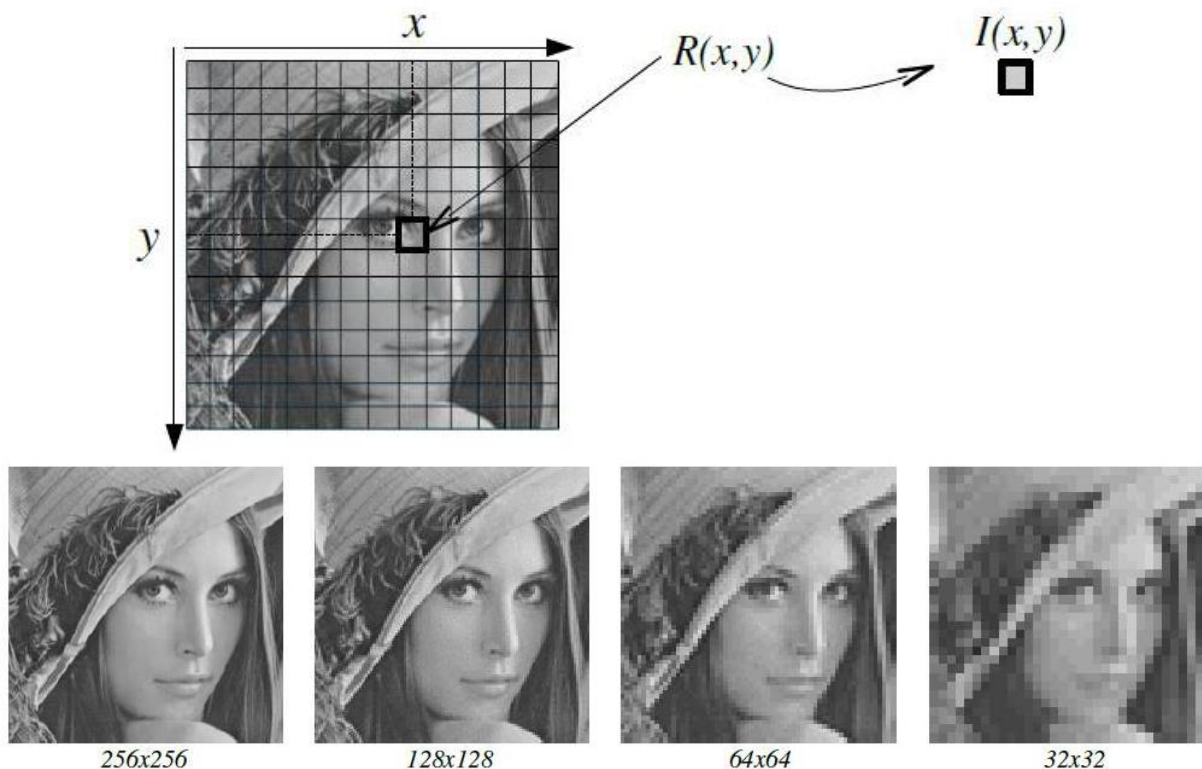
But : transformer un signal analogique en un signal numérique. Pour ceci, des échantillons ponctuels sont prélevés régulièrement sur le signal, évidemment, tout le reste est perdu.

La fréquence à laquelle on prélève les échantillons (fréquence d'échantillonnage) doit être bien choisie :

- Assez grande pour pouvoir bien restituer le signal.
- Pas trop grande pour limiter l'espace de stockage nécessaire.

Lors de la numérisation, on cherche à conserver une qualité maximale (un nombre de pixels maximal, un nombre de couleurs maximal) et à obtenir des données les moins volumineuses possibles. Le problème est que ces deux besoins sont antagonistes. Il conviendra alors de chercher la résolution et le nombre des niveaux de couleurs satisfaisants les besoins. Par exemple, si le but est uniquement la visualisation sur un moniteur donné, il est inutile d'échantillonner l'image à une résolution supérieure à celle du moniteur.

Nous pouvons affirmer alors que l'échantillonnage désigne le procédé de discrétisation spatiale d'une image consistant à associer à chaque zone rectangulaire  $R(x,y)$  d'une image continue une unique valeur  $I(x,y)$  comme mentionné sur les figures 1.9. Notons que la taille d'un pixel est reliée à l'échantillonnage.



**Figure 1.10 :** Échantillonnage d'une image

## 6.4 La quantification

La quantification désigne le choix du bon nombre de bits pour coder les images numériques comme nous pouvons le constater sur la figure 1.10.

But : affecter une valeur numérique à chaque échantillon prélevé sur le signal (après échantillonnage) tout en réduisant le nombre de bits nécessaires au codage.

L'une dépend du capteur, et de sa capacité effective à observer des signaux de valeurs différentes : le rapport signal sur bruit.

Le rapport signal sur bruit est défini à partir du rapport entre l'amplitude des niveaux de gris mesurables par le capteur ( $n_{\max} - n_{\min}$ ) et le niveau du bruit, en gros l'écart type  $s_n$  de la perturbation aléatoire qui affecte les niveaux de gris. En prenant le logarithme, on a le nombre de bits utile au capteur pour coder les images. Outre les capacités du capteur, le nombre de bits réellement nécessaires pour coder une image varie d'une image à l'autre, en fonction de leur contenu informationnel.

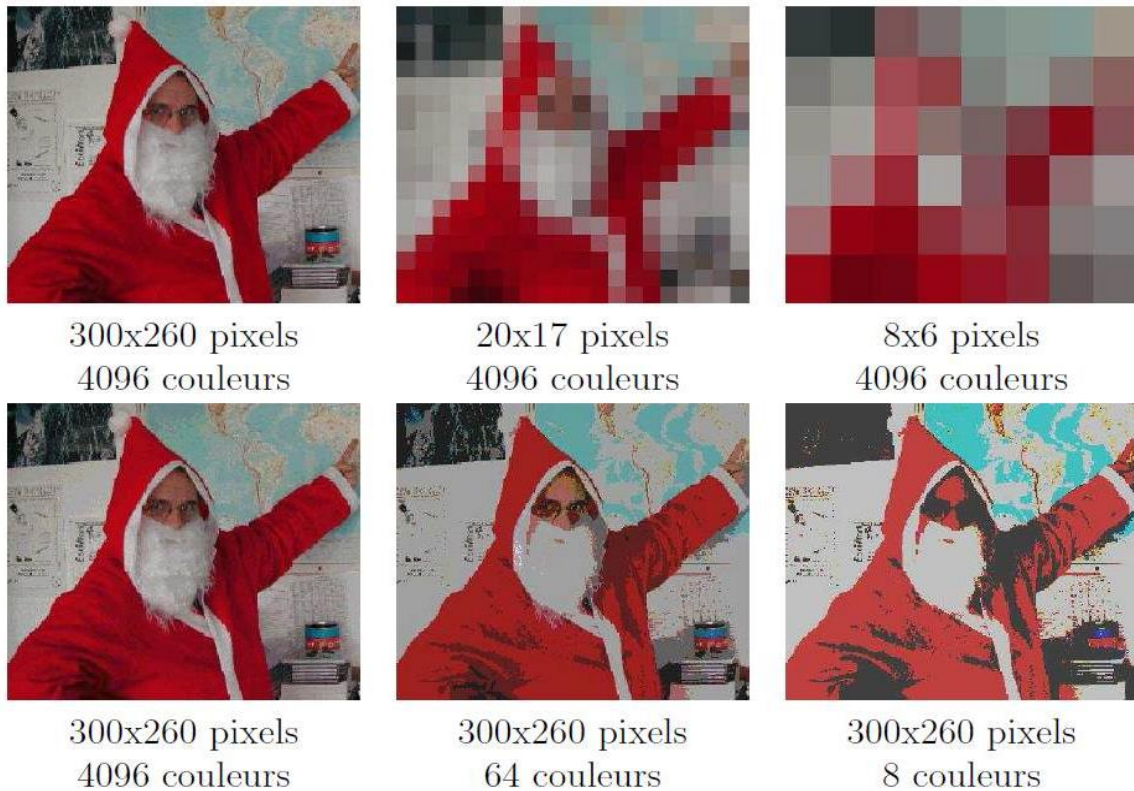
Ce nombre dépend de l'entropie [Cover & Thomas, 2006], définie à partir de la distribution des niveaux de gris de l'image.



**Figure 1.10 :** Quantification d'une image

L'échantillonnage et la quantification ne sont pas limités à la numérisation d'une image. Ce sont des techniques que l'on utilise également sur des images déjà numérisées, afin de modifier la résolution (on ré-échantillonne l'image) ou le nombre de couleurs utilisés (quantification). Cependant, il est dans ce cas fait appel à des techniques de reconstruction d'images, c'est-à-dire de retrouver un signal 2D continu en espace et/ou en couleur.

Nous présentons sur la figure ci-dessous l'échantillonnage et la quantification d'un signal 2d continu selon différentes résolutions spatiales et colorimétriques.



**Figure 1.11** : l'échantillonnage et la quantification d'un signal 2d continu selon différentes résolutions spatiales et colorimétriques

Donc, nous pouvons affirmer que la résolution spatiale est obtenue après échantillonnage (Sampling), déterminant le plus petit détail perceptible dans l'image, en effet, la taille des CCD et le pas d'échantillonnage de l'image au niveau de la carte de numérisation définissent la résolution spatiale de l'image. Le format de l'image est défini par l'entête du fichier et peut être précisé par son extension .bmp, .tif, .gif, .jpeg etc...., il renseigne sur le mode de présentation des données et leur degré de compression [Sayood, 2017]. Tandis que la résolution en niveaux de gris est obtenue après quantification déterminant le plus petit changement de niveau de gris discernable dans l'image.

## 6.5 Stockage des images

Pour adapter le tableau de valeurs entières désignant le code génétique de l'image aux supports et moyens de transmission informatiques, nous devons le retranscrire en binaire. Les infos qui vont être stockées sont la largeur et la hauteur ainsi que les valeurs des pixels. Nous pouvons stocker davantage d'informations telles que le type de données, l'auteur, la date, les conditions d'acquisition, ... etc. Nous allons ainsi stocker les informations concernant l'image dans un en-tête puis les données. Les données sont souvent stockées dans l'ordre des pixels de gauche à droite et de haut en bas en remplaçant chaque valeur entière de niveau de gris par son code en binaire, i.e. sa valeur en base 2 comme mentionné sur la figure I.14.



La ligne suivante 640 480 précise les dimensions de l'image. Viennent ensuite les données, ligne par ligne et pour chaque ligne de gauche à droite.

## Chapitre II : Analyse de Fourier, convolution, filtrage

Les prétraitements ont pour but de faciliter la segmentation d'image en renforçant la ressemblance entre pixel appartenant à une même région, ou en accentuant la dissemblance entre pixels appartenant à des régions différentes. Les images comportent généralement une grande quantité de données. Pour éviter des temps de traitement trop élevés, nous cherchons à leur appliquer des opérateurs locaux, qui ne font appel, pour le calcul qu'à un nombre limité de pixels, situés dans un proche voisinage du pixel courant.

Dans la suite, nous présentons une introduction aux outils d'analyse d'images numériques les plus courants. Pour des raisons didactiques, la présentation est organisée selon les principaux modèles mathématiques qui ont été employés pour traiter les images. Aux différents modèles présentés correspondent des outils fondamentaux, qui se sont révélés au cours du temps plus ou moins incontournables, que ce soit d'un point de vue pratique ou théorique. Citons : la convolution, la transformée de Fourier, l'histogramme, les pyramides, la corrélation, la transformée en tout ou rien, les ondelettes. Nous donnons dans la suite de ce cours une introduction aux outils les plus basiques ou les plus courants.

### 1- L'histogramme

#### 1.1 Définition

Nous entendons par l'histogramme d'une image  $f$  : la fonction discrète  $H$  qui associe à chaque valeur d'intensité le nombre de pixels prenant cette valeur c'est-à-dire  $H$  traduit le nombre d'apparitions du niveau de gris  $x$  dans l'image  $f$ .

$$Hf: \{0, \dots, 255\} \rightarrow \mathbb{N}$$
$$x \rightarrow \text{Card}\{(x, y) | f(x, y) = x\}$$

Si  $f$  est de taille  $M \times N$  à valeurs dans  $\{0, \dots, 255\}$ ,

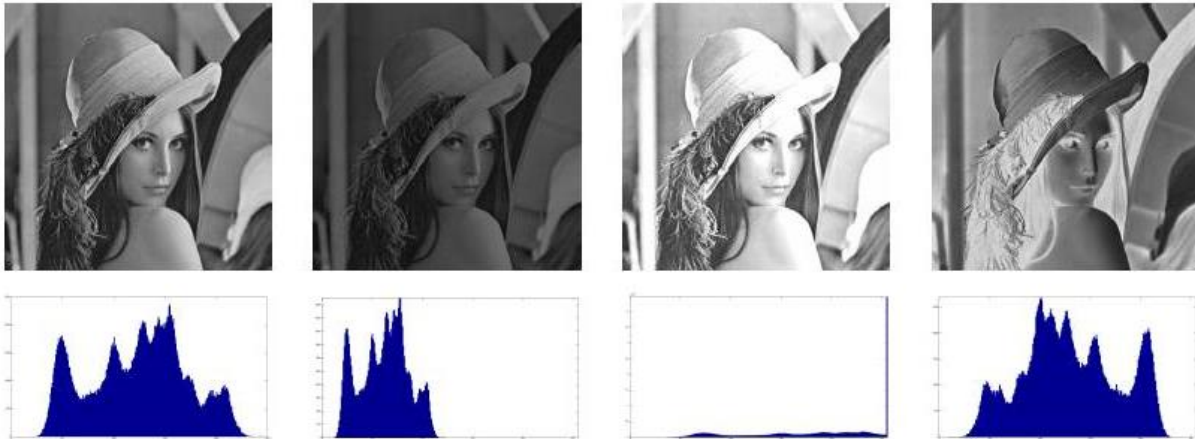
$$\sum_x Hf(x) = MN$$

L'histogramme est un outil privilégié en analyse d'images car il représente un résumé simple, mais souvent suffisant du contenu de l'image : pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.

Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée.

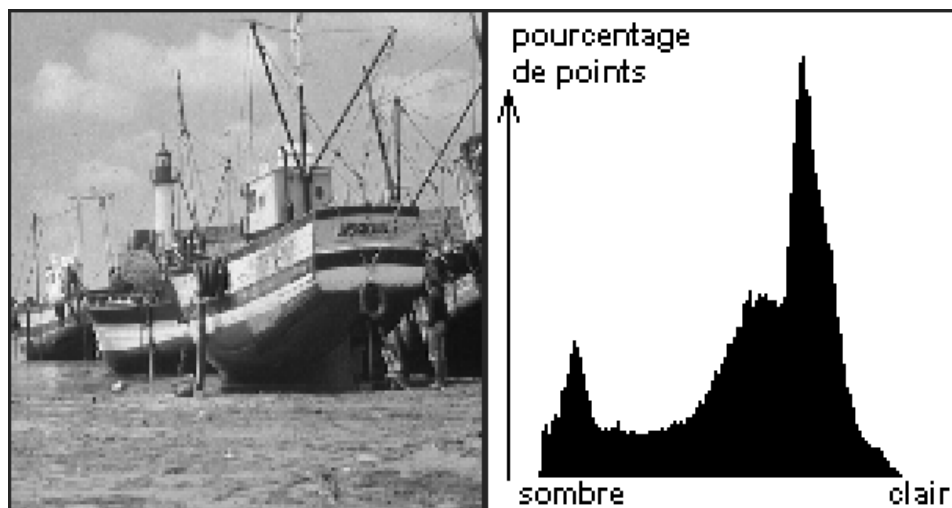
Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

Nous retiendrons cependant qu'il ne faut pas considérer l'histogramme comme une caractéristique fondamentale de l'image dans la mesure où nous pouvons le transformer radicalement sans changer significativement l'image. Il est à noter que l'histogramme ne représente pas la répartition spatiale : deux images peuvent posséder le même histogramme sans pour autant se ressembler. Nous présentons sur la figure 2.1, les histogrammes relatifs à diverses images. En abscisse, nous représentons les niveaux de gris et en ordonnées leurs fréquences d'apparition. L'histogramme des niveaux de gris nous informe sur la concentration de l'image.



**Figure 2.1 :** Histogrammes relatifs à diverses images

L'histogramme d'une image fournit des Information sur la répartition des intensités : moyenne, variance, énergie, entropie, contraste, illumination. Ses modes correspondent à des objets d'intérêt comme il est mentionné sur la figure ci-dessous (nuages, parties du bateau, etc.)



**Figure 2.2 :** Histogramme d'une image comportant des objets d'intérêt séparés du fond

Nous distinguons également l'histogramme cumulé  $H_c(x)$  qui représente le nombre de pixel de l'image possédant un niveau de gris inférieur ou égal à celui du pixel (x). Cet histogramme se calcule par la formule suivante :

$$H_c(x) = \sum_{j=0}^x H(j)$$

## 1.2 Opérations sur l'histogramme

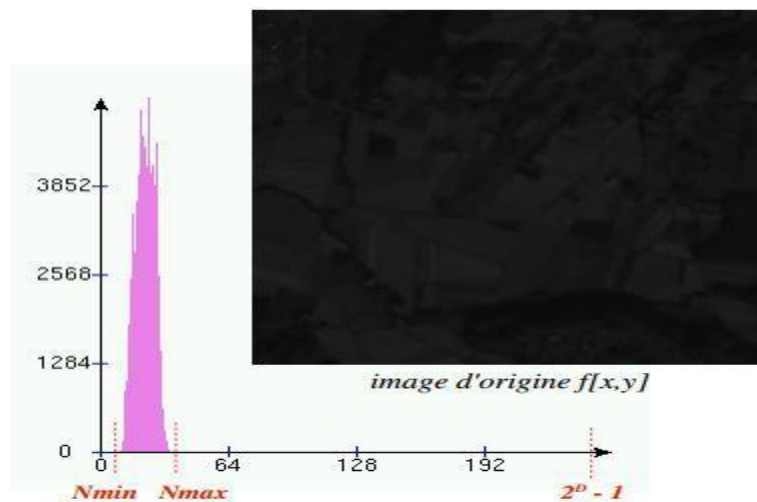
- **Normalisation/Expansion de la Dynamique**

La normalisation d'histogramme, ou expansion de dynamique, est une transformation affine du niveau de gris des pixels de telle sorte que l'image utilise toute la dynamique de représentation.

$$f_{new}[x, y] = (f[x, y] - N_{min}) \frac{255}{N_{max} - N_{min}}$$

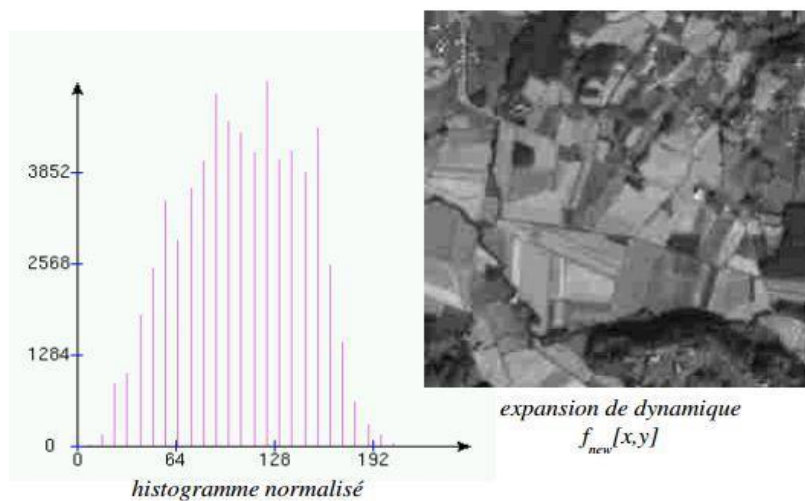
Où  $N_{min}$  présente la plus petite valeur dans l'image et  $N_{max}$  la plus grande valeur.

Prenons l'exemple de l'image  $f$  ci-dessous n'utilisant pas l'ensemble de la dynamique possible :



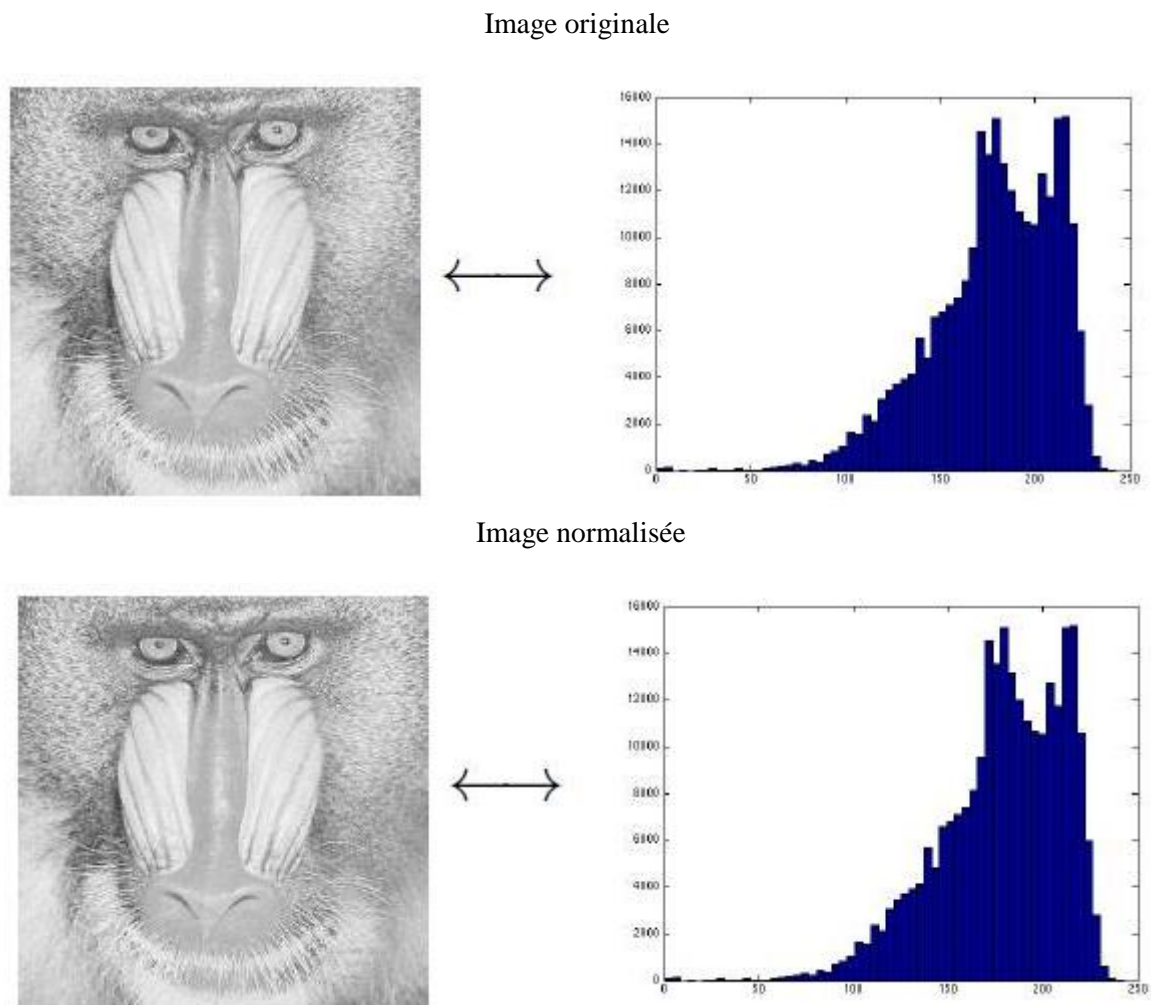
**Figure 2.3 :** Image n'utilisant pas la totalité de la dynamique

Nous allons étirer ou normaliser l'histogramme comme mentionné sur la figure 2.4 :



**Figure 2.4 :** Expansion de dynamique de l'image de la figure 2.3

Nous devons mentionner que dans le cas où l'histogramme initial occupe déjà toute la plage de dynamique, aucun changement n'est visible. La figure 2.5 l'atteste :



**Figure 2.5 :** absence du changement de l'histogramme suite à la normalisation

- **Égalisation d'histogramme**

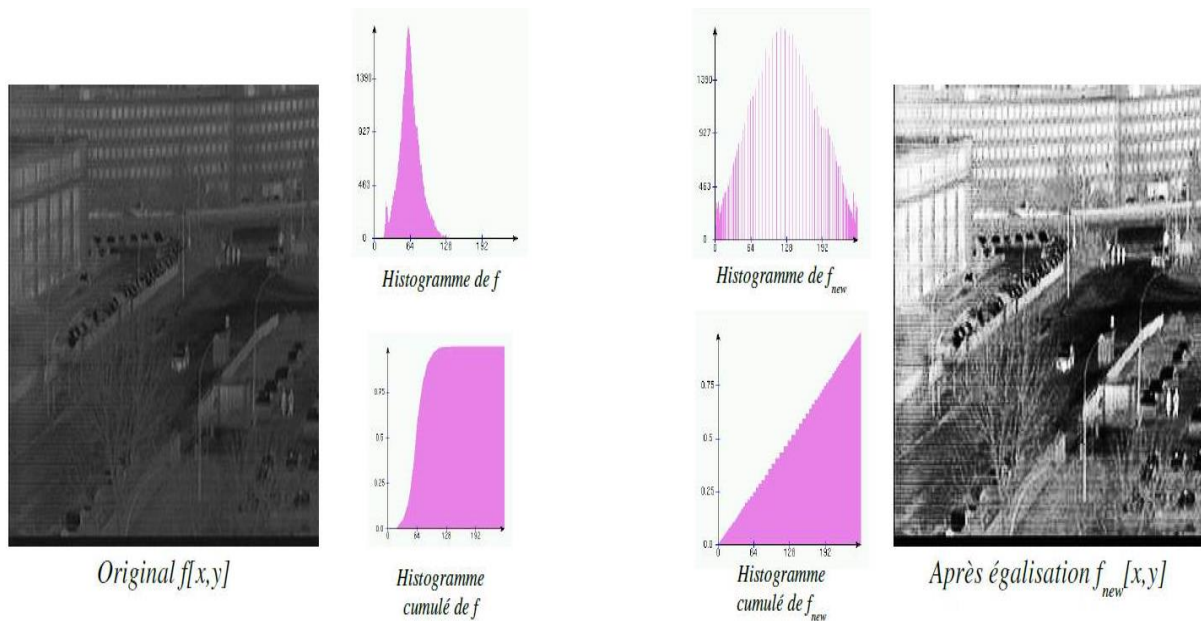
L'égalisation d'histogramme est une transformation des niveaux de gris dont le principe est d'équilibrer le mieux possible la distribution des pixels dans la dynamique (Idéalement, on cherche à obtenir un histogramme plat). En effet, cela consiste en un ajustement automatique du contraste pour étaler les valeurs de l'image afin de les mieux répartir sur l'ensemble des valeurs possibles. Ceci, en allouant plus de niveaux d'intensités dans les zones hautes de l'histogramme et vice-versa.

La technique classique consiste à rendre « le plus linéaire possible » l'histogramme cumulé de l'image en utilisant la transformation suivante :

$$f_{new}[x, y] = (2^D - 1) \frac{HC(f[x, y])}{wh}$$

Où D présente la dynamique, (w,h) la dimension de l'image et HC(.) l'histogramme cumulé.

Nous présentons sur la figure suivante un exemple sur l'égalisation d'histogramme d'une image.



**Figure 2.6 :** Égalisation d'histogramme d'une image

Nous pouvons constater que le résultat est une augmentation globale du contraste dans l'image.

## 2- La transformée de Fourier

### 2.1 Représentation spatiale des images 2D

Une image est une répartition d'intensités lumineuses dans un plan, donc un signal à deux dimensions. Elle peut être continue (représentée par une fonction  $f(x,y)$  continue) ou discrète, c'est alors un tableau ou une matrice de nombres : c'est le cas en des images numériques traitées par ordinateur.

- **Représentation continue**

$$D \in \mathbb{R}^n \rightarrow \mathbb{C}$$

Où  $D$  est un domaine borné,  $\mathbb{C}$  est un espace de couleurs et  $n$  est la dimension, si  $n = 2$  : image usuelle (projection d'une scène 3D sur un plan).

- **Représentation discrète (image numérique)**

$$I : D \subset \mathbb{Z}^n \rightarrow [0 \dots M_i] \times \dots \times [0 \dots M_k]$$

$D$  : domaine fini (support discret)

$n$  : dimension de l'image

$k$  : nombre de bandes ou de canaux : usuellement  $k=1$  (monochrome) ou  $k=3$  (couleur)

$M_i$  : nombre de niveaux par canal

$$I : D \subset \mathbb{Z}^n \rightarrow \underbrace{[0 \dots M] \times \dots \times [0 \dots M]}_k [0 \dots M]^k$$

Le plus souvent :

$$M = 2^8 : [0 \dots 255]^3 = 16\,777\,216 \text{ (8bits par canal)}$$

$$M = 2^{12} : [0 \dots 4096]^3 = 68\,266\,491\,904 \text{ (12bits par canal)}$$

## 2.2 Représentation fréquentielle d'une image 2D

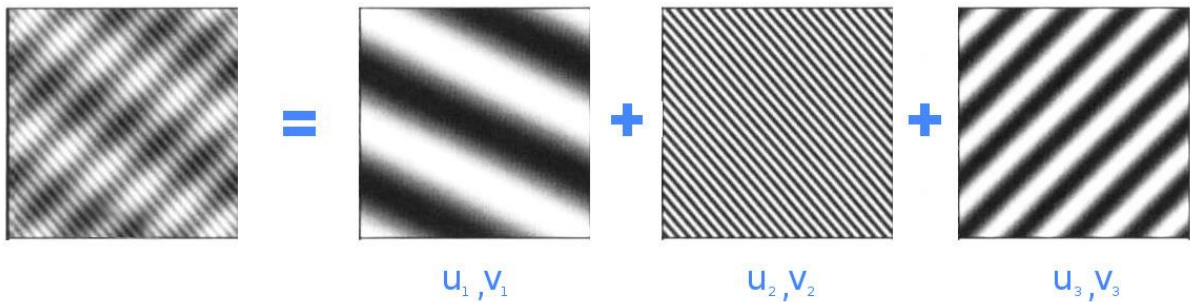
Les questions suivantes sont en ordre :

Il s'agit de quoi ? pourquoi ? Et comment ?

- Il s'agit de quoi ? : c'est un changement d'espace de représentation.

La fonction  $f(x, y)$  décrit une image dans le domaine **spatial**.

$f$  est réécrite comme une somme **d'oscillations**.



Chaque oscillation est caractérisée par un vecteur  $(u, v)$ .

Nous décrivons la contribution de chaque oscillation au moyen d'une fonction notée  $F(u, v)$ .

Changement de représentation :  $f(x, y) \rightarrow F(u, v)$ .

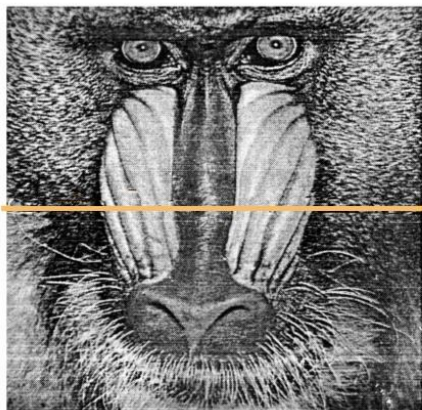
Possible quelle que soit l'image à représenter.

- Pourquoi ? : pour exploiter les caractéristiques fréquentielles.

**Fréquences basses** : transitions lentes, zones homogènes...

**Fréquences hautes** : transitions rapides, détails...

**Composante continue** : centre de l'image (fréquence zéro = moyenne de l'image)



Permet d'effectuer des **traitements dans le domaine fréquentiel** :

- Analyse des caractéristiques de l'image.
- Traitements restreints à certaines fréquences.
- Suppression de défauts périodiques.

- Comment ? : transformée de Fourier.

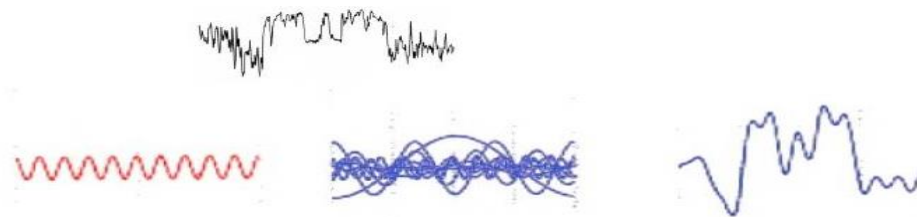
C'est un outil fondamental d'analyse en traitement du signal, le pendant bidimensionnel de la TF et sa version discrète peut être appliqué avec profit aux images numériques.

Nous distinguons divers domaines d'application :

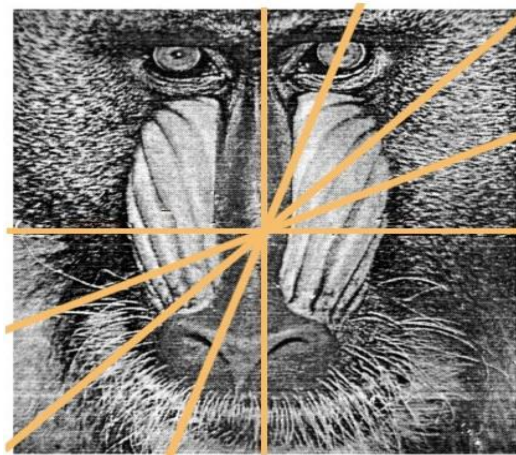
- Filtrage : lissage, augmentation de la netteté.
- Restauration : élimination des dégradations.
- Classification : distinction de différents types d'images.
- Compression : utilisée, mais certaines transformées sont plus adaptées (transformée en z).

La transformée de Fourier Discrète (DFT) est un outil très utilisé en traitement d'images car il permet d'observer les caractéristiques fréquentielles des images [Oppenheim & Schaffer, 2010 ; Bracewell, 2000].

Le modèle fréquentiel tend à décrire l'image en termes de structures périodiques, comme une décomposition en une somme de fréquence, par exemple des oscillations sinusoïdales.



En 2D, prise en compte des transitions dans toutes les directions.



Chaque fréquence directionnelle correspond à une oscillation  $(u, v)$ .

La transformée de Fourier permet la décomposition d'un signal image  $f$  en combinaison linéaire de sinusoïdes complexes.

La transformée de Fourier de l'image  $f(x,y)$ , de largeur  $N$  et de hauteur  $M$  est donnée par :

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

Où  $f(x,y)$  est l'intensité du pixel à la position  $(x, y)$ ,  $0 \leq x \leq M-1$ ,  $0 \leq y \leq N-1$ .

$(x, y)$  sont les coordonnées du domaine spatial,  $(u, v)$  sont les coordonnées du domaine fréquentiel dans les directions  $x$  et  $y$ .

Les coefficients  $F(u, v)$  dit coefficients de Fourier, fournissent des informations sur les fréquences  $(u,v)$  et permettent des manipulations dans le domaine fréquentiel.

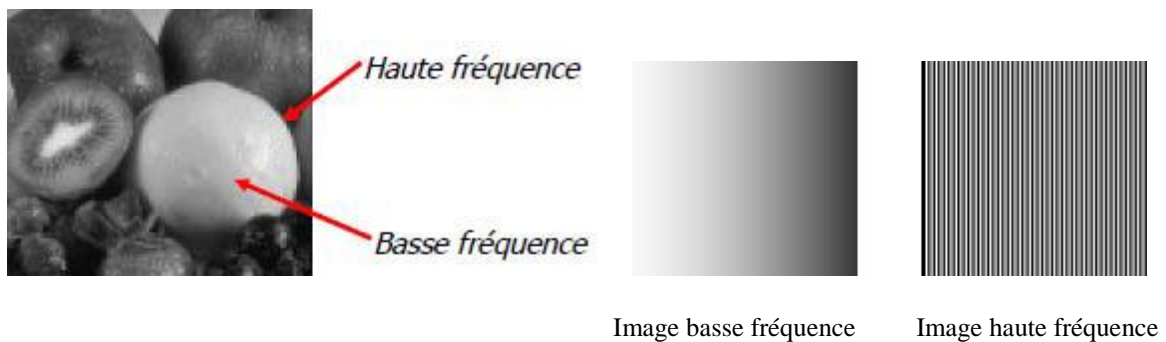
A cet effet, nous obtenons une nouvelle représentation de l'image dans l'espace des fréquences :

- Un signal est une somme de sinusoïdes à différentes fréquences.
- Une image réelle est une superposition de sinusoïdes 2D.

Nous calculons la TF en chaque pixel de l'image pour obtenir une image complexe. À une image sinusoïdale 2D correspond un point dans l'espace des fréquences. Pour visualiser cette image, nous représentons sa norme (spectre d'amplitude) et sa phase.

Les valeurs de l'amplitude en général présentent une très grande dynamique (les basses fréquences sont plus importantes que les hautes fréquences), le spectre d'amplitude représente le logarithme du module de la transformée de Fourier.

**La fréquence dans une image représente la variation de l'intensité des pixels de l'image** comme indiqué sur la figure 2.7, les basses fréquences (correspondent à des changements d'intensité lents) représentent les régions homogènes et floues, tandis que les hautes fréquences (correspondent à des changements d'intensité rapides) représentent les contours et les changements brusques d'intensité.



**Figure 2.7 :** Haute fréquence et basse fréquence d'une image.

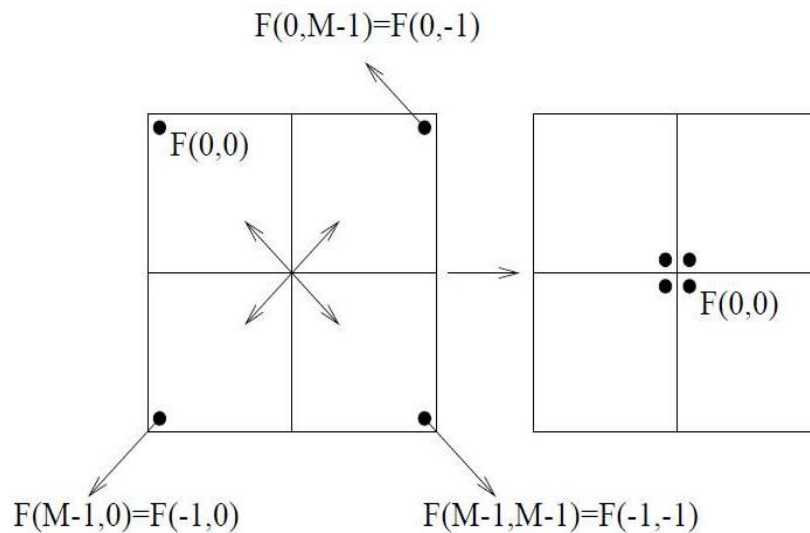
### 2.3 Interprétation de la transformée de Fourier discrète

Que pouvons-nous voir dans une image de la transformée de Fourier ? Nous y voyons le spectre de l'image. Nous présentons sur la figure 2.8, l'image Lena ainsi que sa transformée de Fourier discrète.



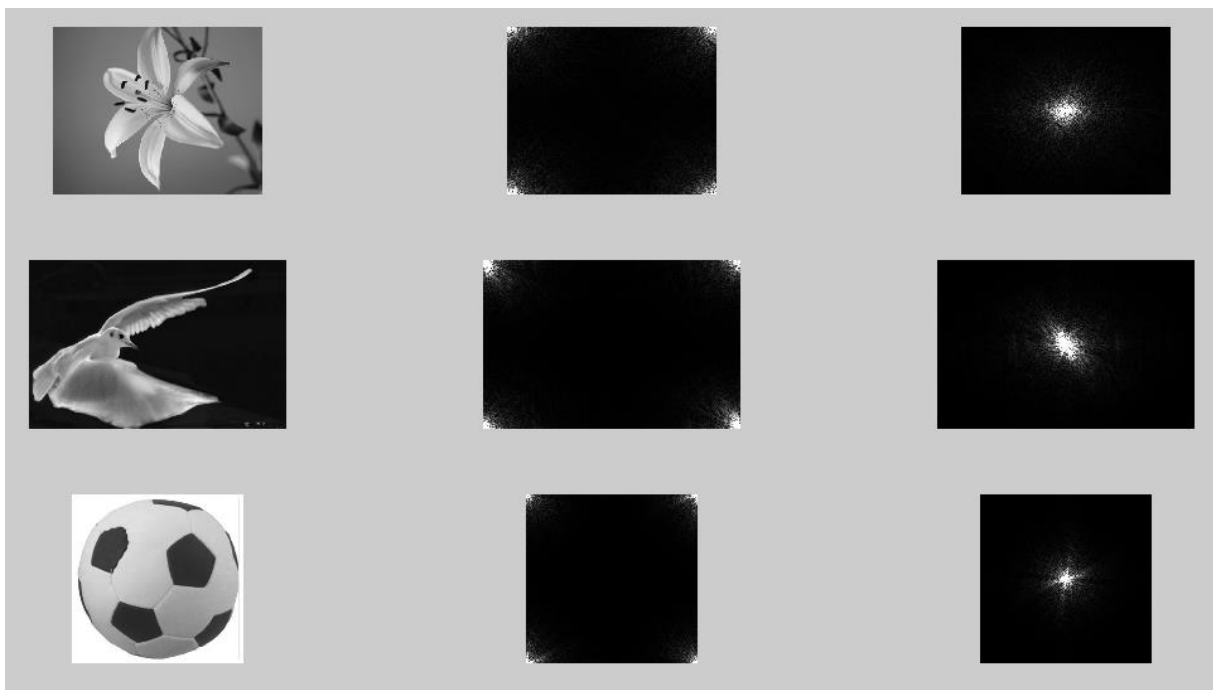
**Figure 2.8 :** L'image Lena et le module de sa transformée de Fourier discrète.

Une première remarque que nous pouvons faire concernant la visualisation de la transformée de Fourier discrète, est que la composante fréquentielle  $F(0,0)$ , appelée composante continue ou composante DC, se trouve en haut et à gauche de l'image alors que nous avons l'habitude de voir cette composante située au milieu de l'image. En utilisant les propriétés de périodicité, il est possible d'observer le module de la transformée de Fourier d'une manière plus conventionnelle comme mentionné sur la figure 2.9.



**Figure 2.9:** Décalage du spectre pour arriver à centrer l'origine.

Il est plus naturel de voir l'origine des fréquences au centre de l'image du spectre. Cela ne change pas l'information contenue dans le spectre, seul son agencement est modifié. La figure suivante démontre des spectres modifiés par décalage de l'image.



**Figure 2.10 :** Spectres modifiés par décalage de l'image.

Le spectre de Fourier varie sensiblement d'une image à une autre. Néanmoins, la plupart des spectres d'images présentent des caractéristiques communes telles que la présence de droites, d'une épaisseur plus ou moins importantes qui passent par l'origine.

- Le spectre informe sur la présence de certaines fréquences (et de leur importance) dans l'image.
- Plus une droite est longue, plus elle porte des fréquences élevées.
- La direction des droites indique les lignes de force de l'image d'origine, qui lui sont perpendiculaires.
- Si une droite est constituée de points ou tries alignés, cela indique une certaine périodicité de l'image dans la direction perpendiculaire.

Nous commençons par l'analyse des images ne contenant que la texture synthétisée afin d'éviter de voir des fréquences parasites interférer avec les fréquences de la texture d'étude. Si la texture possède une certaine structure, il en ira de même pour le module de la transformée de Fourier : si la texture est moins ordonnée, voire complètement aléatoire, le spectre du module de la transformée de Fourier de cette texture sera également assez aléatoire et nous ne saurons rien en tirer. Néanmoins, le spectre de la TF présente généralement certains pics. Ces pics nous informent sur l'orientation de la structure ainsi que sur son amplitude. A savoir :

- Le pic dominant dans le spectre donne la direction principale de la texture.
- La localisation des pics permet de déterminer la période spatiale fondamentale de la texture.
- Chaque pic représente une périodicité spatiale dans un sens particulier.

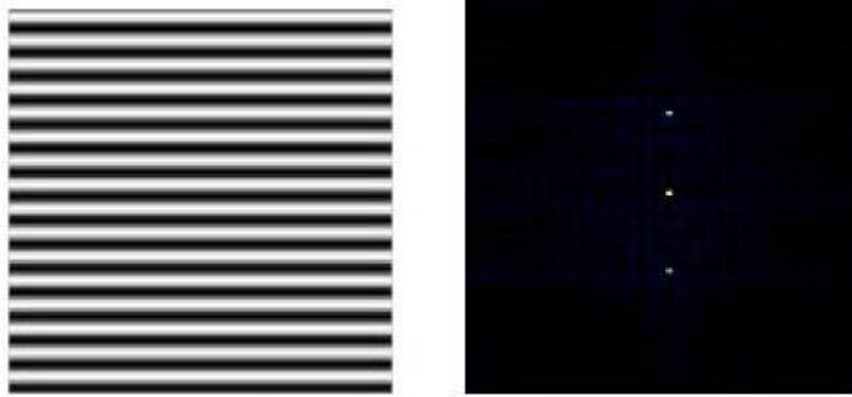
Les images qui sont de purs cosinus ou de purs sinus ont des transformées de Fourier particulièrement simples.

La figure 2.11 représente un sinus horizontal. Nous distinguons un motif de direction verticale qui se répète dans l'image. Sa transformée de Fourier présente trois points alignés horizontalement, donc dans la direction perpendiculaire à celle du motif.



**Figure 2.11** : Spectre d'une image de sinus horizontal.

Quant à la figure 2.12, elle représente un sinus vertical. Sa transformée de Fourier présente aussi trois points alignés, mais verticalement, c'est à dire dans la direction perpendiculaire à celle du motif.



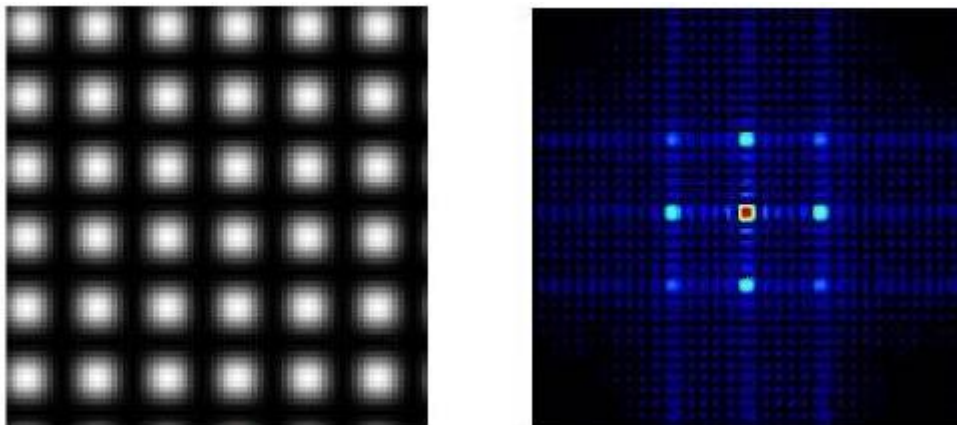
**Figure 2.12 :** Spectre d'une image de sinus vertical.

Nous pouvons affirmer que dans les deux cas, la transformée de Fourier est constituée de trois points :

- Le point central (de coordonnées  $(0,0)$ ) représente la moyenne de l'intensité de l'image initiale.
- Les deux autres points représentent la fréquence (verticale ou horizontale) des sinus.

Nous remarquons aussi que plus la fréquence est élevée plus les points sont espacés. De plus, la première image dont on prend la transformée est invariante selon l'axe des  $y$ . C'est pour cela que la transformée de Fourier ne peut être non nulle que pour  $v = 0$ . De même pour la deuxième image qui est invariante sur l'axe des  $x$ .

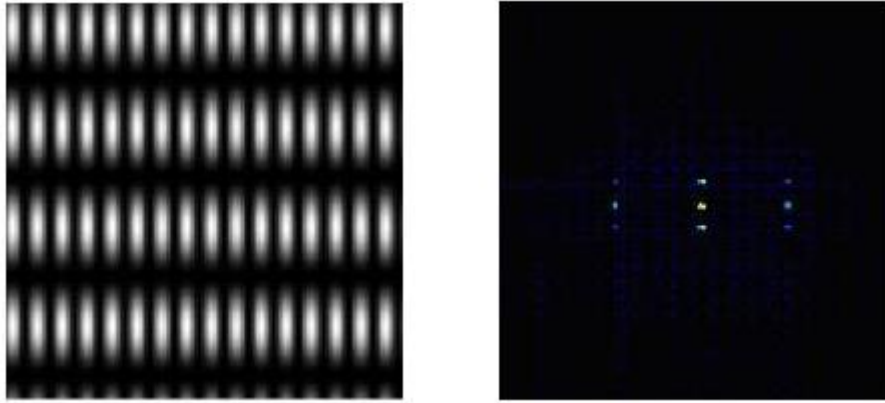
Nous allons évoquer le cas d'une image constituée de sinus dans les deux directions  $x$  et  $y$ , nous obtenons la figure 2.13 :



**Figure 2.13 :** Même périodicité horizontale et verticale.

Nous avons obtenu 9 points en dehors du point central. Ces points représentent la périodicité dans le sens horizontal, vertical et diagonal. Les autres points d'intensité plus faible sont dus à des effets de bord.

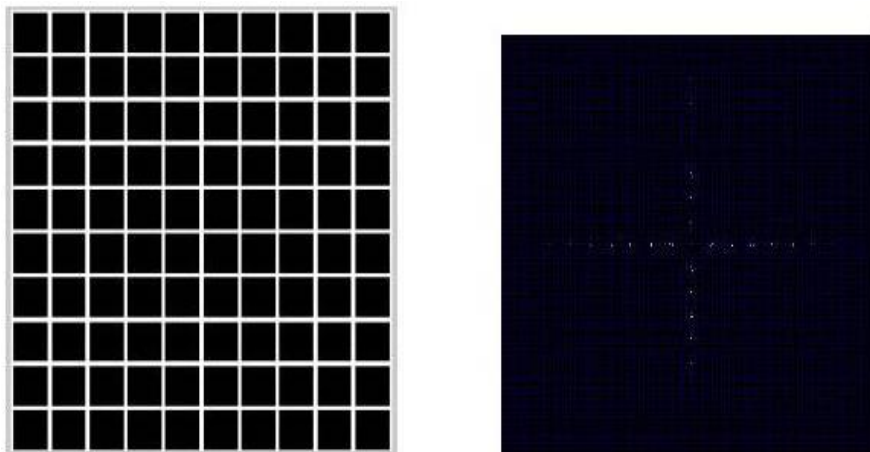
Le lien entre l'espacement des pics du spectre et la périodicité du motif dans l'image est mis en évidence dans la figure 2.14, qui présente des sinus avec deux fréquences différent horizontalement et verticalement :



**Figure 2.14 :** Périodicités horizontale et verticale différentes.

En effet, les points sont plus espacés horizontalement car la fréquence horizontale est plus élevée que la fréquence verticale.

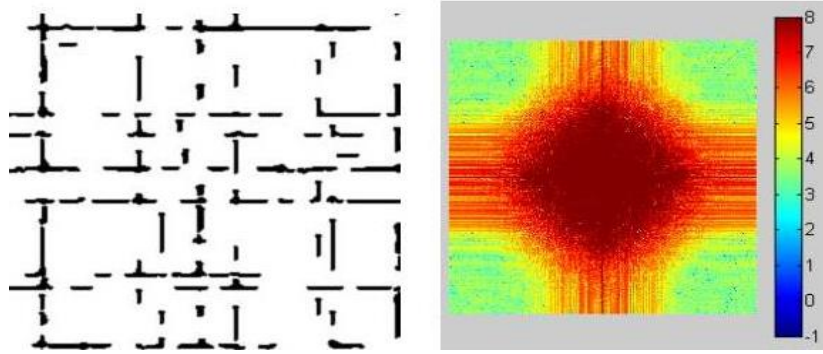
Nous notons dans les exemples précédents que lorsqu'il y a dans l'image d'origine des lignes fortes, apparaissent dans sa transformée de Fourier des lignes perpendiculaires à ces lignes fortes. Ceci se retrouve dans d'autres exemples comme indiqué sur la figure 2.15 :



**Figure 2.15 :** TF d'une image de lignes perpendiculaires.

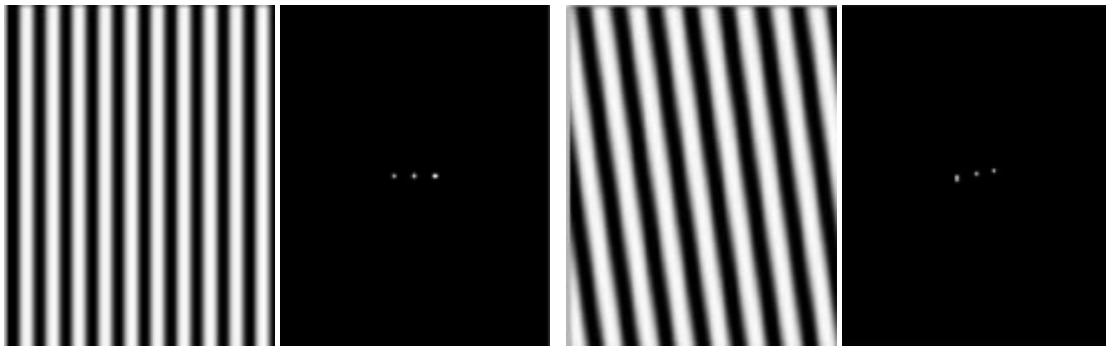
Cette image est composée de carrés répartis avec une certaine périodicité selon les composantes horizontales et verticales. Aux différentes lignes horizontales correspondent dans la transformée de Fourier les points de la ligne verticale, qui est une ligne dominante. Aux différentes lignes verticales correspondent dans la transformée de Fourier les points de la ligne horizontale. Les périodicités verticales et horizontales des lignes se retrouvent dans l'espacement régulier des points dans la transformée de Fourier.

Nous présentons sur la figure 2.16 le spectre d'une image de texture réelle présentant des lignes perpendiculaires. Nous pouvons affirmer que le spectre est, certes, moins démonstratif que dans le cas d'une texture synthétique. Néanmoins, on retrouve une concentration de l'énergie autour des deux axes principaux. La présence d'une large zone blanche homogène au niveau de l'image initiale explique la dominance des basses fréquences au niveau du spectre.



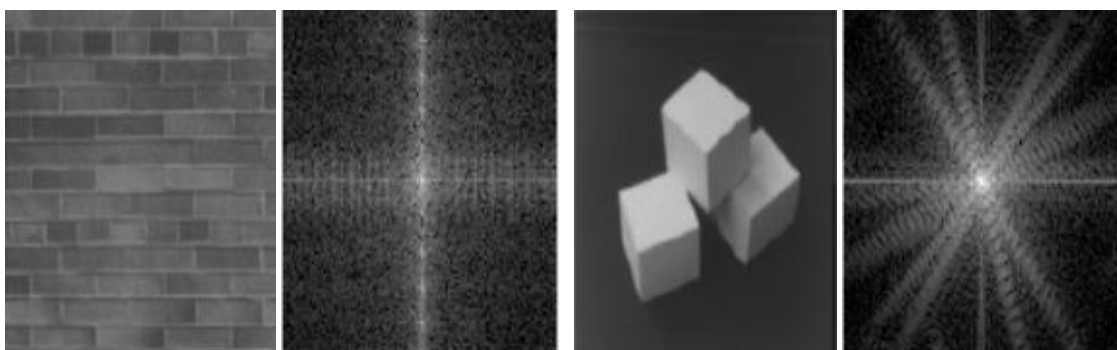
**Figure 2.16 :** Spectre d'une texture présentant des lignes perpendiculaires.

Nous allons analyser sur la figure 2.17 le spectre d'une image formée de sinusôides dans la direction horizontale, ceci, dans le cas avant et après que nous appliquons une rotation sur l'image.



**Figure 2.17 :** Image sinusoidale et son spectre à gauche ainsi qu'une image avec une rotation et son spectre à droite.

Nous pouvons affirmer que ceci engendre une rotation de la TF avec le même angle. Sur la figure 2.18, nous évoquons d'autres cas :



**Figure 2.18 :** Image possédant des lignes horizontales/verticale à gauche ainsi qu'une image possédant des lignes dans toutes les directions à droite.

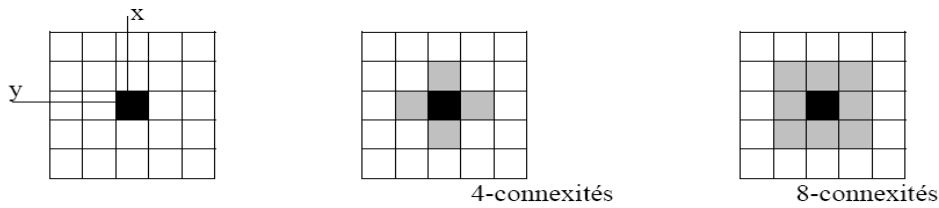
Nous pouvons constater que sur l'image de gauche qui possède des lignes horizontales/verticales, ceci est bien retrouvé dans sa transformée, idem pour l'image de droite qui possède des lignes dans toutes les directions, ceci est aussi retrouvé dans sa transformée.

### 3- La convolution

#### 3.1 Topologie de l'image

L'image est représentée par un ensemble de pixels déposés sous la forme d'une grille [Gonzalez & Woods, 2018].

$$\begin{array}{|c|c|c|} \hline I(x-1, y-1) & I(x, y-1) & I(x+1, y-1) \\ \hline I(x-1, y) & I(x, y) & I(x+1, y) \\ \hline I(x-1, y+1) & I(x, y+1) & I(x+1, y+1) \\ \hline \end{array}$$



#### 3.2 Définition et calcul d'une convolution

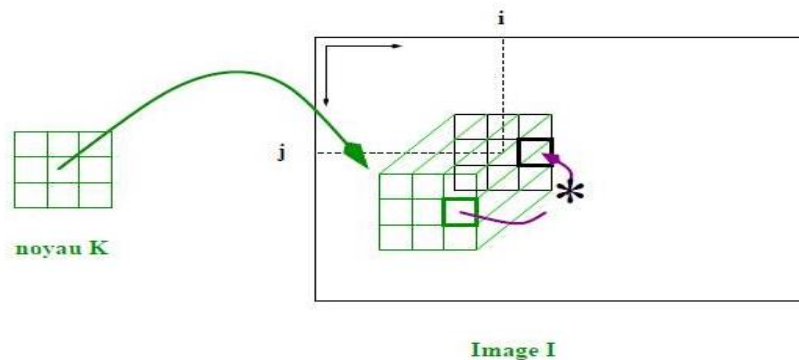
C'est l'opérateur de base du traitement linéaire des images. Apparue très tôt dans les premiers systèmes d'analyse d'images sous forme empirique et justifiée par des considérations d'implantation, ce n'est que plus tard qu'on a fourni des justifications physiques et fait le lien théorique avec les filtres et le traitement du signal.

Pour calculer une convolution, on remplace la valeur de chaque pixel par la valeur du produit scalaire entre les valeurs du noyau de convolution et les valeurs du voisinage du pixel considéré.

Soit une image  $I_1$  de dimension finie que nous désirons lui appliquer une convolution  $K$  de dimension  $3 \times 3$  ( $K$  pour kernel), les pixels de l'image  $I_2$  obtenue de  $I_1$  par  $K$  :

$$I_2(i, j) = \sum_{k=0}^2 \sum_{l=0}^2 I_1(i+k-1, j+l-1) K(k, l)$$

Une illustration de ceci est donnée sur la figure suivante :



**Figure 2.19 :** Réaliser la convolution d'une image  $I$  par un masque de convolution  $K$  revient à remplacer chaque pixel  $(i, j)$  par la somme des produits des éléments du masque avec les pixels correspondants, en centrant le masque sur le pixel  $(i, j)$ .

Pour donner davantage de détails sur la manipulation du masque de convolution, la figure 2.20 complète la précédente. Sachant que le noyau K est le suivant :

$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline -1 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

Noyau K

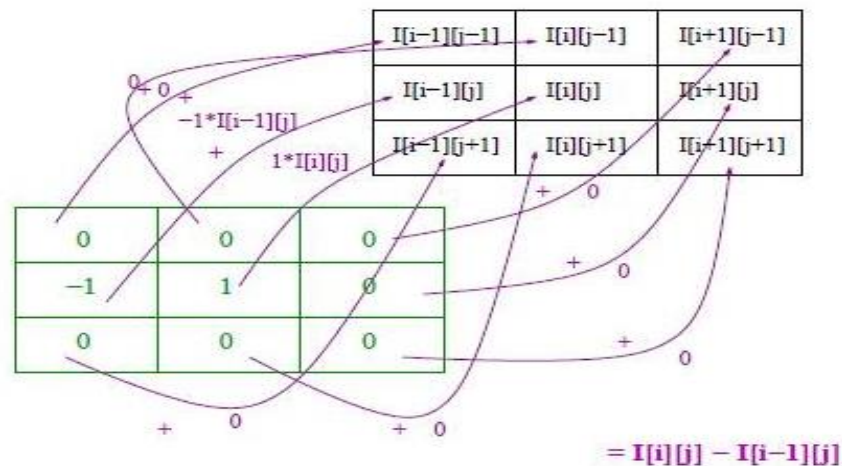


Figure 2.20 : Convolution par le noyau K, exemple du pixel (i, j)

## 4- Filtrage d'image

Les systèmes d'acquisition comme un scanner ou une caméra produisent un ensemble d'images ; ces dernières ne sont pas toujours parfaites et elles présentent généralement des distorsions (sous la forme de bruit d'image) nuisibles pour une visualisation optimale.

Nous allons évoquer des techniques d'amélioration des images numériques, pour augmenter la qualité de leur rendu visuel, ou pour faciliter leur analyse. Nous cherchons alors à atténuer, sinon supprimer une certaine dégradation. Celle-ci n'est pas forcément connue a priori, mais elle peut parfois être estimée a posteriori. Nous distinguons :

- Les dégradations liées au bruit :  $g(x) = f(x)+b(x)$  ou  $g(x) = f(x)b(x)$  liées au capteur, à la quantification, à la transmission...etc. Nous donnons recours à ces lacunes par le filtrage, consistant à appliquer une transformation (appelée filtre) à toute ou partie d'une image numérique en utilisant un opérateur. Suite au filtrage, l'intensité d'un pixel est transformée en fonction des intensités sur un petit voisinage du pixel.

### 4.1 Définition d'un filtre

Un filtre est une transformation mathématique (appelée produit de convolution) permettant, pour chaque pixel de la zone à laquelle il s'applique, de modifier sa valeur en fonction des valeurs des pixels avoisinants, affectées de coefficients. Par différenciation, les techniques de filtrage permettent en outre de calculer ou amplifier les contrastes locaux.

Le filtre est représenté par un tableau (matrice), caractérisé par ses dimensions et ses coefficients, dont le centre correspond au pixel concerné. Les coefficients du tableau déterminent les propriétés du filtre. Nous donnons un exemple de filtre 3 x 3 coefficients :

1	1	1
1	4	1
1	1	1

Ainsi le produit de la matrice image, généralement très grande car représentant l'image initiale (tableau de pixels), par le filtre donne une matrice correspondant à l'image traitée.

## 4.2 Notion de bruit

Le bruit caractérise les parasites ou les interférences d'un signal, c'est-à-dire les parties du signal déformées localement. Ainsi le bruit d'une image désigne les pixels de l'image dont l'intensité est très différente de celles des pixels voisins.

Le bruit peut provenir de différentes causes :

- Environnement lors de l'acquisition.
- Qualité du capteur.
- Qualité de l'échantillonnage.

Nous distinguons divers types de bruit :

- a) **Bruit d'acquisition, de numérisation, de transmission** : les incertitudes dans les différentes étapes de formation de l'image numérique induisent des fluctuations aléatoires de la valeur des pixels (bruit gaussien). Les erreurs de transmission font apparaître des valeurs aberrantes (bruit impulsionnel).



**Figure 2.21** : A droite exemple d'un bruit gaussien et à gauche un autre d'un bruit impulsionnel

- b) **Bruit de compression** : les techniques de compression d'image avec perte produisent une distorsion dans l'image, comme cet effet de bloc (figure 2.22) dans la transformée Jpeg (taux de compression 1/25).



**Figure 2.22 :** Bruit de compression Jpeg.

- c) **Bruit spatial fixe** : la non-uniformité des détecteurs dans la matrice de l'image infra-rouge de la figure 2.23 entraîne une texturisation de l'image.



**Figure 2.23 :** Bruit spatial fixe.

### 4.3 Les types de filtres

Nous distinguons les types de filtres suivants :

- a) **Les filtres passe-bas** : consistant à atténuer les composantes de l'image ayant une fréquence haute (pixels foncés). Ce type de filtrage est généralement utilisé pour atténuer le bruit de l'image, c'est la raison pour laquelle nous parlons habituellement de lissage (filtre anti-bruit). En effet, les filtres de lissage sont des opérateurs qui éliminent des éléments perturbateurs / non significatifs dans les images numériques, soit pour améliorer leur visualisation, soit pour les simplifier en but d'un traitement postérieur [Smith, 2007].
- b) **Les filtres passe-haut** : à l'inverse des passe-bas, atténuent les composantes de basse fréquence de l'image et permettent notamment d'accentuer les détails et le contraste, c'est la raison pour laquelle le terme de " filtre d'accentuation " est parfois utilisé. L'accentuation (ou bruitage) est l'inverse du lissage ; il s'agit d'une opération visant à accentuer les différences entre les pixels voisins. Ainsi l'accentuation peut permettre de mettre en exergue les limites entre les zones homogènes de l'image et est alors appelée extraction de contours (également contourage ou réhaussement de contours). Les filtres passe-bande permettant d'obtenir la différence entre l'image originale et celle obtenue par application d'un filtre passe-bas.

## 4.4 Filtrage dans le domaine spatial

Nous entendons par filtrage spatial, un filtrage qui s'applique sur un voisinage d'un pixel dans une image. En effet, le concept de filtrage spatial consiste à déplacer un masque de convolution de pixel en pixel afin de parcourir toute l'image. La réponse au filtre est calculée en chaque point de l'image en utilisant une relation prédéfinie.

Si le masque de convolution est un masque linéaire (nous parlons alors de filtrage linéaire), la réponse est donnée par la somme des valeurs des pixels du voisinage, chacune de ces valeurs étant pondérée par le coefficient du masque lui correspondant.

Si le filtre spatial est non linéaire (nous parlons de filtrage non linéaire), la réponse donnée n'est pas une somme pondérée mais un résultat découlant d'une relation prédéfinie (comme le filtre médian qui remplace la valeur d'un pixel par la valeur médiane des pixels qui l'entourent).

Nous évoquerons un ensemble de filtres passe-haut, dont certains effectueront un filtrage linéaire et d'autres effectueront un filtrage non linéaire :

- Le filtre moyenneur, gaussien et exponentiel comme filtres linéaires passe-bas, et le filtre médian comme filtres non linéaires passe-bas.
- Les filtres réhausseur, de Prewitt, de Sobel et de Laplace comme filtres linéaires passe-haut.

**4.4.1 Filtre moyenneur :** appelé également mean filtering, son principe est simple : un pixel est remplacé par la moyenne de lui-même ainsi que ses voisins. Les filtres diffèrent selon la définition du voisinage. Nous pouvons considérer un voisinage en connexité 4 ou 8 ou même plus large.

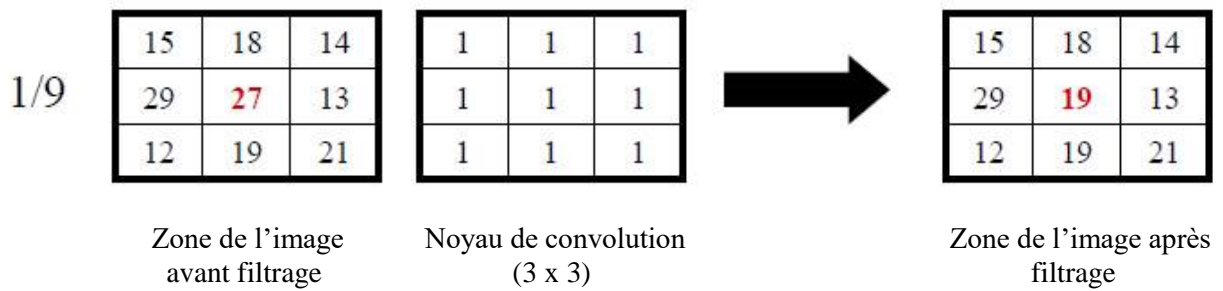
$$\begin{bmatrix} 0 & \frac{1}{5} & 0 \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ 0 & \frac{1}{5} & 0 \end{bmatrix} \quad \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

**Connexité 4**

**Connexité 8**

Nous pouvons affirmer que les filtres moyenneurs sont un type de filtres passe-bas dont le principe est de faire la moyenne des valeurs des pixels avoisinants. Le résultat de ce filtre est une image plus floue. Le filtre moyenneur fait disparaître des transitions brutales d'intensité dans l'image. Or les bruits aléatoires sont typiquement caractérisés par des transitions brutales d'intensité, ce qui rend cette technique particulièrement appropriée pour l'élimination de tels bruits. Cependant les contours des objets et les bords se trouvant dans une image sont aussi caractérisés par de fortes variations locales en intensité. L'application d'un filtre moyenneur tend donc à faire disparaître ou du moins à atténuer ces caractéristiques de l'image, ce qui peut produire un effet de flou non souhaité.

Nous exposons sur la figure 2.24 un exemple d'application d'un filtre moyenneur sur une partie d'une image en montrant le changement de la valeur du pixel centré.



**Figure 2.24 :** Application d'un filtre moyenneur sur une partie d'une image.

Quant à la figure 2.25, nous présentons une image avant et après que nous lui avons appliqué un bruit blanc.



**Figure 2.25 :** A gauche l'image originale et à droite la même image bruitée par un bruit blanc.

Nous allons démontrer sur la figure 2.26 les résultats du filtrage moyenneur de l'image bruitée présentée sur la figure précédente.



**Figure 2.26 :** A gauche l'image filtrée par un noyau 3x3 et à droite l'image filtrée par un noyau 5x5

Nous attestons que le filtrage permet d'éliminer une partie du bruit, notamment dans la partie du ciel mais que les détails sont altérés, notamment sur les maisons et le paysage. En effet, le filtre moyenneur est un filtre passe-bas permettant ainsi d'éliminer les hautes fréquences, correspondants au bruit. Son

inconvenient est qu'il élimine également les hautes fréquences correspondantes aux détails de l'image : il rend ainsi l'image moins bruitée mais plus floue.

**4.4.2 Filtre gaussien :** le filtre gaussien est un filtre linéaire passe-bas, il est principalement utilisé pour rajouter du flou dans une image et pour réduire certains types de bruits. Le principe de ce filtre est d'effectuer une convolution avec une gaussienne [Burger & Burge, 2016].

L'équation d'une gaussienne est la suivante :

$$H(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$$

Le paramètre sigma s'appelle la déviation standard, et détermine la largeur de la cloche Gaussienne. Le noyau d'un filtre gaussien est défini par un ensemble de coefficients qui sont des échantillons de gaussienne 2D. Avec une valeur  $\sigma=0.8$  ( $\sigma$  est l'écart-type), nous obtenons des masques 3x3 et 5x5. La largeur du filtre est donnée par son écart-type  $\sigma$ . Plus  $\sigma$  est grand, plus nous réduisons le bruit avec l'inconvénient d'accentuer le flou dans l'image. Si  $\sigma$  est trop grand, tous les détails de l'image sont perdus.

Etant donné que les images numériques sont composées de valeurs discrètes, le noyau gaussien sera composé aussi de valeurs discrètes :

H(-1, -1)	H(0, -1)	H(1, -1)
H(-1, 0)	H(0, 0)	H(1, 0)
H(-1, 1)	H(0, 1)	H(1, 1)

$$H = \frac{1}{2\pi\sigma^2} \begin{pmatrix} \exp^{-2/\sigma^2} & \exp^{-1/2\sigma^2} & \exp^{-2/\sigma^2} \\ \exp^{-1/2\sigma^2} & 1 & \exp^{-1/2\sigma^2} \\ \exp^{-2/\sigma^2} & \exp^{-1/2\sigma^2} & \exp^{-2/\sigma^2} \end{pmatrix}$$

Si  $\sigma=0.8$  par exemple, nous avons le filtre 3 x 3 suivant :

$$\approx \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Nous divisons par la somme des coefficients 16.

Et le filtre 5x5 suivant :

$$\approx \frac{1}{98} \begin{pmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 6 & 8 & 6 & 2 \\ 3 & 8 & 10 & 8 & 3 \\ 2 & 6 & 8 & 6 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{pmatrix}$$

Nous divisons par la somme des coefficients 98.

L'effet de ce filtre sur l'image est assez similaire au filtre moyeneur, mais la moyenne est pondérée en ce sens où les pixels près du centre ont un effet ou un "poids" plus important que ceux qui sont situés plus loin.

Nous allons démontrer sur la figure 2.27 les résultats du filtrage gaussien sur une image.



Image originale



$\sigma=2$



$\sigma=4$

**Figure 2.27** : Filtrage gaussien d'une image par deux valeurs de  $\sigma$ .

Nous constatons que le filtrage gaussien apporte du flou identiquement au moyeneur, mais il préserve mieux les contours des objets. Cette préservation des contours est moins flagrante si nous utilisons un masque 3x3. Il faut noter que plus  $\sigma$  est grand, plus le flou appliqué à l'image sera marqué.

**4.4.3 Filtre exponentiel** : le filtre exponentiel est un filtre linéaire passe-bas, dit aussi filtre de lissage. Il est principalement utilisé pour rajouter du flou dans une image et pour réduire certains types de bruits. A partir de l'équation suivante d'une courbe exponentielle :

$$H(x, y) = \frac{\gamma^2}{4} \exp -\gamma(|x| + |y|)$$

Le noyau d'un filtre exponentiel est défini par un ensemble de coefficients qui sont des échantillons de la courbe exponentielle 2D. Avec une valeur de  $\gamma = 0.8$ , nous obtenons des masques 3x3 et 5x5.

$$H = \frac{\gamma^2}{4} \begin{pmatrix} \exp^{-2\gamma} & \exp^{-\gamma} & \exp^{-2\gamma} \\ \exp^{-\gamma} & 1 & \exp^{-\gamma} \\ \exp^{-2\gamma} & \exp^{-\gamma} & \exp^{-2\gamma} \end{pmatrix}$$

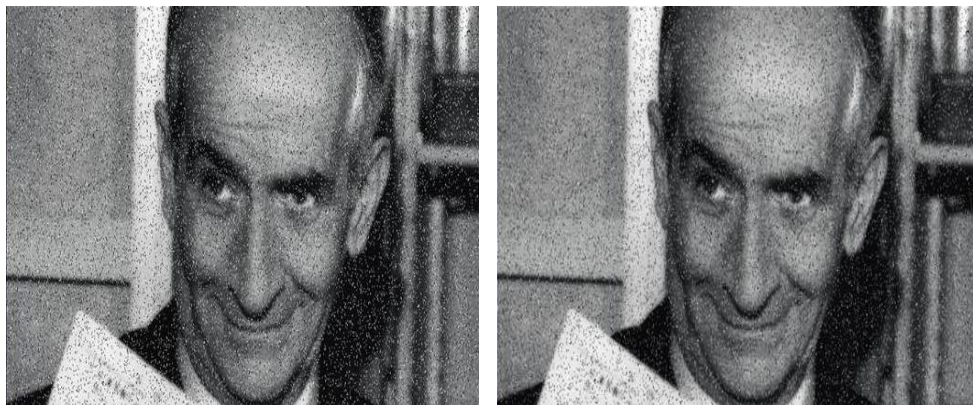
Nous avons le filtre 3x3 suivant :

$$\mathbb{R} \frac{1}{32} \begin{array}{|c|c|c|} \hline 1 & 3 & 1 \\ \hline 3 & 16 & 3 \\ \hline 1 & 3 & 1 \\ \hline \end{array}$$

Ainsi que le filtre 5x5 suivant :

$$\mathbb{R} \frac{1}{98} \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 3 & 1 & 1 \\ \hline 1 & 3 & 7 & 3 & 1 \\ \hline 3 & 7 & 16 & 7 & 3 \\ \hline 1 & 3 & 7 & 3 & 1 \\ \hline 1 & 1 & 3 & 1 & 1 \\ \hline \end{array}$$

Nous allons démontrer sur la figure 2.28, les résultats du filtrage exponentiel de l'image bruitée par un bruit poivre et sel.



**Figure 2.28** : A droite l'image bruitée par un bruit poivre et sel et à droite l'image filtrée par un filtre exponentiel.

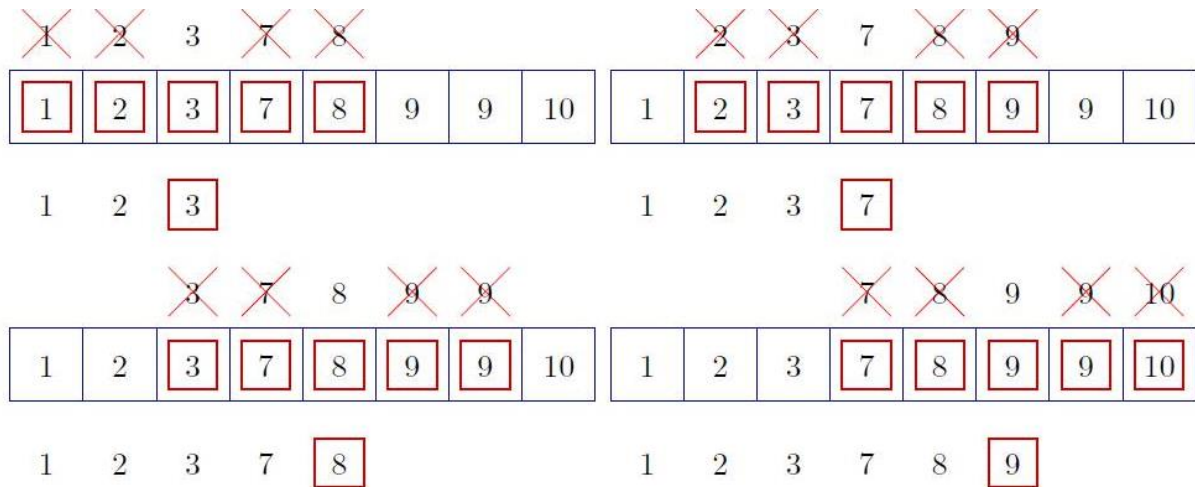
Nous pouvons affirmer que ya pas vraiment de différences sensibles entre le filtre exponentiel et le filtre gaussien. Mais les conclusions relatives au filtre gaussien s'appliquent aussi au filtre exponentiel vis-à-vis du filtre moyeneur.

**4.4.4 Filtre médian** : Le principe du filtre médian consiste à remplacer un pixel par la médiane de ses voisins. En effet, nous choisissons la valeur médiane dans un voisinage :

$$g(x) = \text{med}\{f(z) \mid z \in S_x\}$$

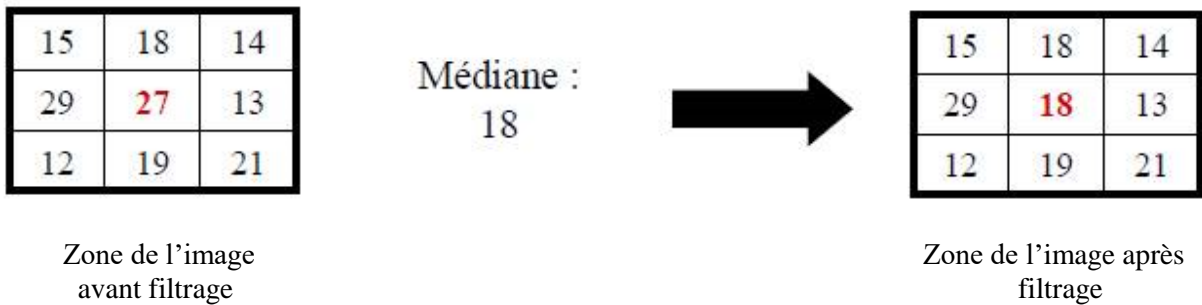
$S_x$  étant une fenêtre centrée en  $x$ , souvent carrée de taille fixée.

Ainsi, même si plusieurs pixels voisins sont bruités, nous pouvons corriger le pixel courant. Ce filtre induit cependant un lissage puisque même des pixels corrects peuvent être modifiés. De plus, ce filtrage est plus coûteux car nécessite d'effectuer un tri des voisins pour chaque pixel comme suit :



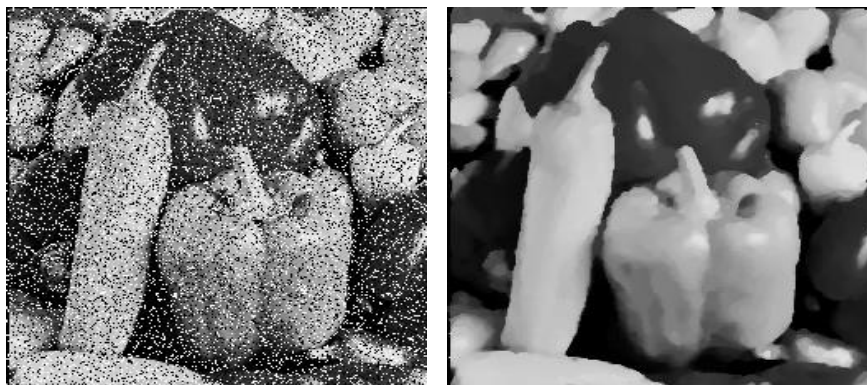
Plus le voisinage considéré est grand, plus l'algorithme sera coûteux. Nous penserons donc, lors de l'implémentation, à utiliser un algorithme de tri rapide tel que le quick sort.

Nous exposons sur la figure 2.27 un exemple d'application d'un filtre médian sur une partie d'une image en montrant le changement de la valeur du pixel centré.



**Figure 2.29 :** Application d'un filtre médian sur une partie d'une image.

Nous allons démontrer sur la figure 2.28 les résultats du filtrage médian d'une image bruitée par un bruit impulsionnel.



**Figure 2.30 :** Bruit impulsionnel traité par un filtre médian.

Nous pouvons constater que le filtrage médian s'avère particulièrement efficace pour juguler les effets d'un bruit impulsif. Sa caractéristique essentielle est sa capacité à conserver des transitions fortes tout en supprimant une partie importante du bruit.

Il a été aussi prouvé que le filtre médian permet d'obtenir de bons résultats sur du bruit poivre et sel comme mentionné sur la figure 2.29 [Pratt, 2007].



**Figure 2.31 :** A gauche image bruitée par un bruit poivre et sel (10%) et à droite image filtrée deux fois par un filtre médian 3x3.

**4.4.5 Filtre réhausseur de contour :** Le filtre réhausseur de contour est un filtre de netteté dit filtre passe-haut, effectuant un filtrage linéaire (par convolution). Le filtre réhausseur que l'on trouve avec différentes valeurs de masques, permet principalement d'accentuer les détails de l'image et d'améliorer les détails qui sont devenus flous suite à une erreur ou à des caractéristiques naturelles du processus d'acquisition employé. Donc, ce filtre peut être utilisé afin de corriger l'effet de lissage des filtres précédents.

Ce type de filtre réhausseur est souvent employé dans le domaine de l'électronique (représentation des circuits électroniques), dans l'imagerie médicale, dans l'imagerie des systèmes militaires. Par conséquence, ce filtre a l'effet fâcheux d'augmenter le bruit.

Nous donnons divers noyaux de convolution :

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

1	-2	1
-2	5	-2
1	-2	1

1	-3	1
-3	9	-3
1	-3	1

**Filtre de Pratt**

**Réhausseur de contraste**

**Figure 2.32 :** Quatre exemples de masques pour des filtres réhausseurs de contours et de contraste

Les deux masque à gauche de la figure sont des masques les plus souvent utilisés, notamment le deuxième qui porte le nom de son inventeur (filtre de Pratt).

Nous présentons sur la figure 2.33 une image bruitée par un bruit poivre et sel (10%) puis lissée deux fois par un filtre médian 3x3. Par la suite, un rehaussement avec le noyau suivant a été réalisé.

0	-1	0
-1	10	-1
0	-1	-1



**Figure 2.33 :** A gauche l'image lissée deux fois par un filtre médian 3x3 et à droite l'image rehaussée.

Nous pouvons constater que le masque réhausseur montre bien les contour réhaussés.

**4.4.6 Filtre de Prewitt et de Sobel :** Ces filtre sont des filtres passe-haut qui sont dédiés à la détection de contours. Leurs particularités est de détecter des lignes horizontales, verticales et diagonales. Comme ils agissent dans le domaine du filtrage linéaire, ils possèdent un masque de convolution. Ces deux filtres s'expriment par la convolution suivante :

<b>Masque de Prewitt horizontal</b>	-1	0	1
	-1	0	1
	-1	0	1

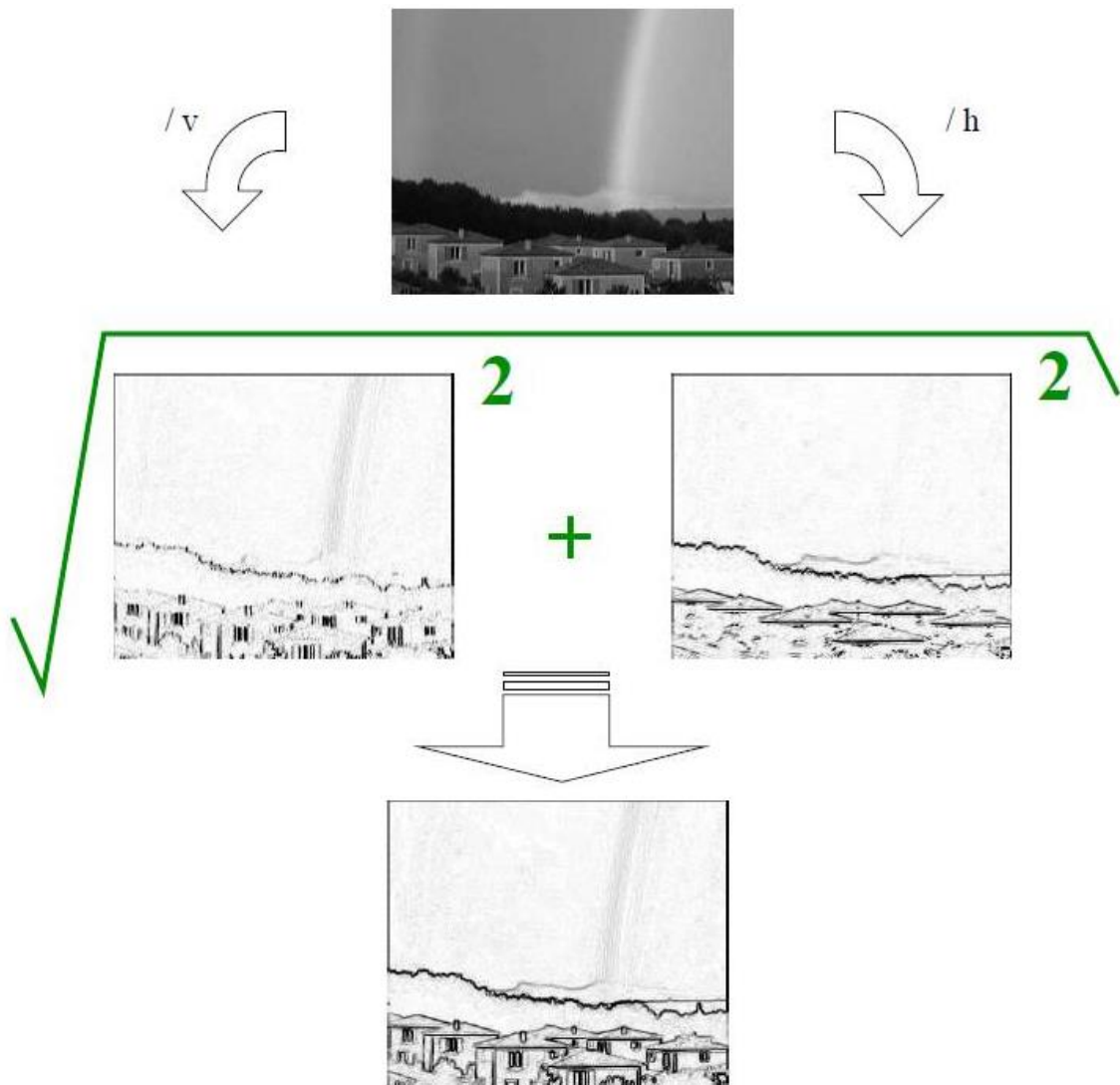
<b>Masque de Prewitt vertical</b>	-1	-1	-1
	0	0	0
	1	1	1

<b>Masque de Sobel horizontal</b>	-1	0	1
	-2	0	2
	-1	0	1

<b>Masque de Sobel vertical</b>	-1	-2	-1
	0	0	0
	1	2	1

Le filtre de Sobel utilise par exemple ses deux noyaux 3x3, l'un pour l'axe horizontal (X) et l'autre pour l'axe vertical (Y).

En effet, le principe consiste à obtenir une première image par le filtre horizontal, puis une deuxième image par le filtre vertical, enfin à obtenir le niveau du pixel en calculant le module des deux premières images. Ce module s'obtient en prenant la racine carrée de la somme des carrés de chaque composante. Une combinaison est effectuée pour créer l'image finale comme mentionné sur la figure 2.34.



**Figure 2.34 :** Application du filtre de Sobel sur une image réelle.

La figure 2.35 filtre une image exemple avec le filtre de Prewitt dans sa direction horizontale. Nous pouvons remarquer que le balayage horizontal fait ressortir les lignes verticales mais que du côté droit. Pour avoir les lignes verticales coté gauche, il faut utiliser ce filtre de facons symétrique. La figure 2.36 filtre une image exemple avec le filtre de Prewitt dans sa direction verticale. Nous pouvons attester que le balayage vertical fait ressortir les lignes horizontales mais que du côté bas. Pour avoir les lignes horizontales coté haut, il faut utiliser ce filtre de facon symétrique.

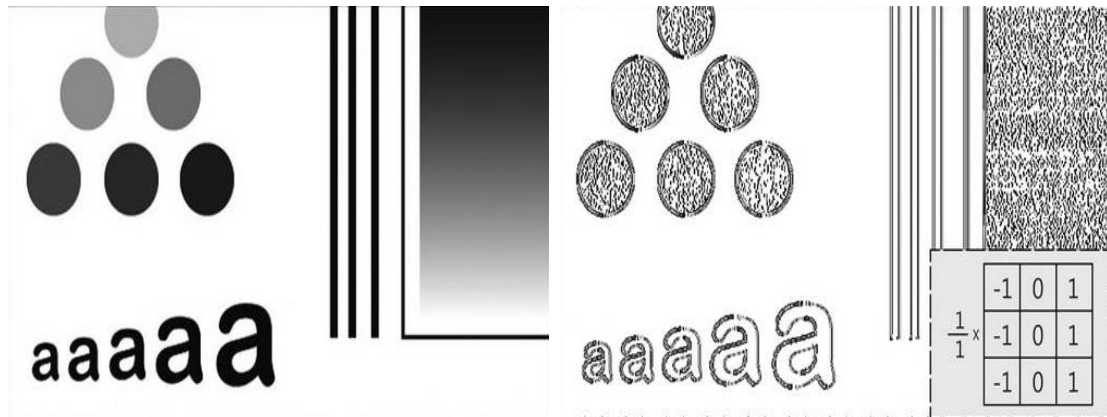


Figure 2.35 : Filtrage d'une image avec le filtre de Prewitt dans sa direction horizontale.

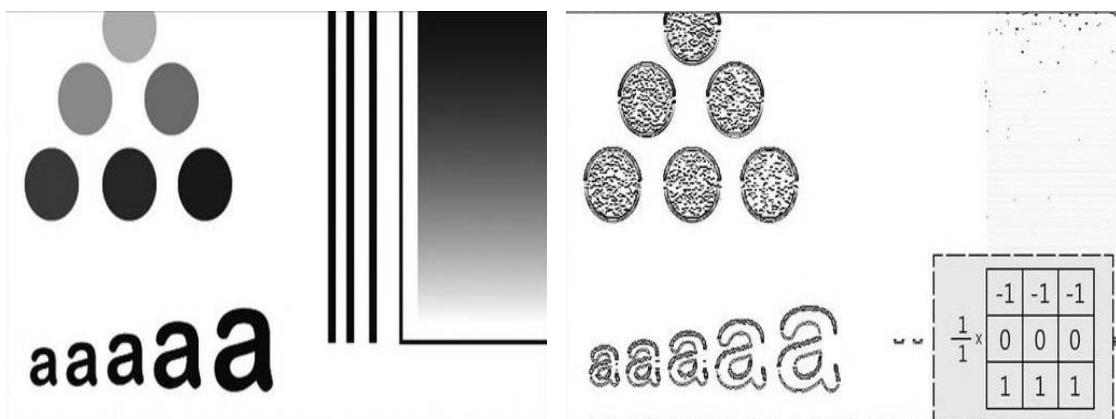


Figure 2.36 : Filtrage d'une image avec le filtre de Prewitt dans sa direction verticale.

**4.4.7 Filtre de Laplace :** Le filtre de Laplace (dit aussi filtre Laplacien) est un filtre passe-haut dédié à la détection de contours. Il est utilisé pour mettre en valeur les détails qui ont une variation rapide de luminosité, donc il fut idéal pour rendre visible les contours des objets. Comme il agit dans le domaine du filtrage linéaire, il possède un masque de convolution qui s'exprime sous la forme d'une connexité 4 et sous la forme d'une connexité 8.

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

1	-2	1
-2	4	-2
1	-2	1

Contrairement au filtre de Prewitt et de Sobel, le filtre Laplacien permet de réaliser la détection de contours en une seule passe. D'où sa très grande utilisation dans le domaine de la détection de contours pour sa souplesse d'application pratique. Son inconvénient majeur est sa très grande sensibilité aux bruits. Les filtres de détection de contours transforment l'image d'entrée en une image noire sauf aux points où un contour est détecté qui est marqué en blanc. En procédant à une inversion de contraste, nous obtenons des contours noirs sur fond blancs. La mise en évidence des contours constitue la différenciation de l'image. Un contour sera bien détecté s'il constitue une frontière entre deux régions homogènes.

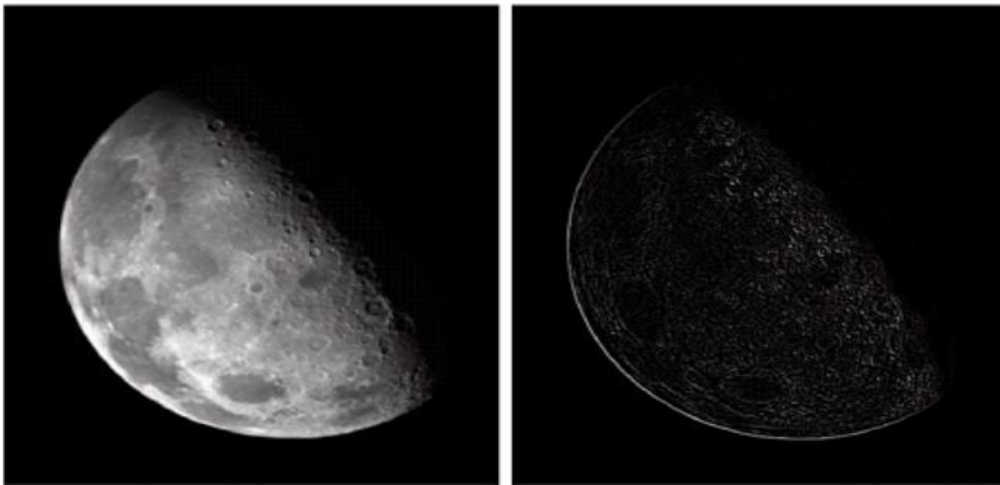
Le contour est un concept essentiellement basé sur la capacité du système visuel humain à distinguer plusieurs zones connexes d'aspect homogène, pour lesquelles la ligne de séparation sera appelée un contour ou une frontière. Il n'y a donc pas de définition précise et nous pouvons dire qu'il y a des contours là où le système visuel humain en voit.

Il a été démontré que le filtre Laplacien donne une bonne détection de contours dans les deux connexités, même si, dans sa connexité 4, l'épaisseur du contour est moins épaisse que celle obtenue dans sa connexité 8. L'opérateur Laplacien est aussi très utilisé dans le domaine de la segmentation.

Un détail est à noter : la somme de tous éléments du noyau d'un filtre Laplacien est toujours nulle, ce qui implique que ce filtre n'est pas un filtre linéaire.

Les propriétés de réduction de bruit des filtres Gaussiens peuvent être utilisées en combinaison avec d'autres filtres qui au contraire génèrent du bruit, comme les filtres Laplaciens. On peut par exemple choisir d'appliquer d'abord un filtre Gaussien pour réduire le bruit, avant d'appliquer un filtre Laplacien pour détecter les points autour desquels les variations de luminosité sont importantes.

Nous présentons sur la figure 2.37 l'application d'un filtre Laplacien sur une image.



**Figure 2.37 :** Filtrage d'une image avec le filtre Laplacien.

Nous pouvons affirmer que la détection de contours est bien effectuée, les lignes courbes qui séparent des zones homogènes sont bien détectées. Le point faible est relatif aux zones foncées et les contours mal délimités se positionnant entre deux régions homogènes.

#### **4.5 Filtrage dans le domaine fréquentiel**

Nous avons évoqué dans la section précédente la notion de filtrage dans le domaine spatial, en effet, les filtres s'écrivaient sous la forme d'un produit de convolution entre une matrice (ou noyau de convolution) et une image. Malheureusement, en appliquant de gros masques de convolution, nous pouvons constater que le temps de calcul devient très important. Pour cela, nous pouvons appliquer les masques dans l'espace des fonctions de Fourier qui rendront les calculs beaucoup plus rapides.

Il est possible d'extraire la composante fréquentielle de l'image avec des filtres passe-haut, passe-bas ou passe band. Un filtrage passe-haut laisse passer les hautes fréquences et atténue les basses fréquences (il a pour effet de faire apparaître les détails de l'image). Un filtrage passe-bas laisse passer les basses fréquences et atténue les hautes fréquences (il a pour effet de faire disparaître les détails de l'image).

Etant donné la propriété de linéarité de la transformée de Fourier, l'image peut être décomposée comme  $IMG=BF+HF$ .

Dans le domaine spatial, le filtrage se fait par convolution, alors que dans le domaine spectral (ou fréquentiel), il se fait par multiplication (ou masquage de l'image).

La transformée de Fourier permet d'explorer la composition fréquentielle de l'image, et de par ses propriétés, de lui appliquer des opérateurs de filtrage. Lors de cette transformation, le signal est décomposé sur un ensemble de signaux de base qui sont cosinus, sinus ou l'exponentielle imaginaire et forment une base orthogonale (qui permet de supprimer les redondances d'informations). Cette technique a été utilisée par plusieurs chercheurs pour tenter de caractériser les images de textures. En effet, elle présente un outil bien utile pour la caractérisation globale de la texture. En outre, l'analyse locale par Fourier n'est pas suffisamment précise et ne renseigne pas sur toutes les informations nécessaires à la caractérisation d'une texture.

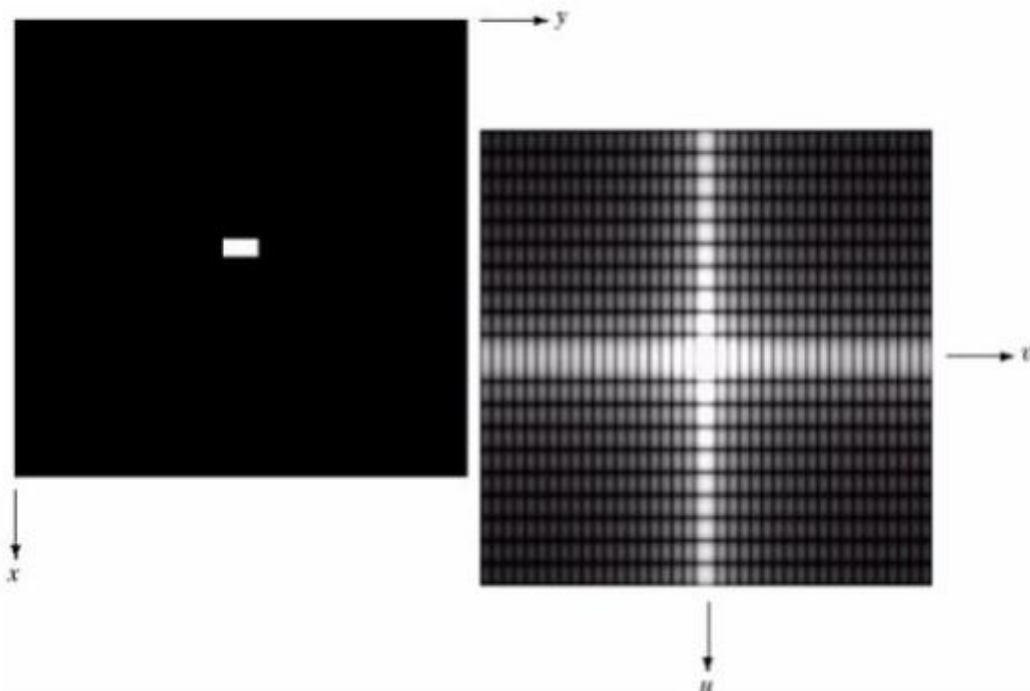
#### 4.5.1 Les étapes du filtrage fréquentiel :

1- Multiplier  $f(x, y)$  par  $(-1)^{x+y}$ .

Nous multiplions la fonction d'entrée par  $(-1)^{x+y}$  pour centrer la fonction transformée.

Selon les propriétés des exposants :

$$TF [(-1)^{x+y} f(x, y)] = F(u - \frac{M}{2}, v - \frac{N}{2})$$



2- Calculer la transformée de Fourier de  $f \Rightarrow F(u, v)$ .

Interprétation intuitive du spectre :

- Chaque terme de  $F(u, v)$  est fonction de toutes les valeurs de  $f(x, y)$  pondérées par l'exposant.
- Les fréquences décrites par le spectre de la transformée de Fourier sont reliées directement aux taux de changements de niveaux de gris dans l'image.
- La valeur de  $F(0, 0)$  est la moyenne, à une fréquence nulle.
- Plus nous nous éloignons du centre, plus la fréquence augmente.

3- Multiplier  $F(u, v)$  par un filtre.

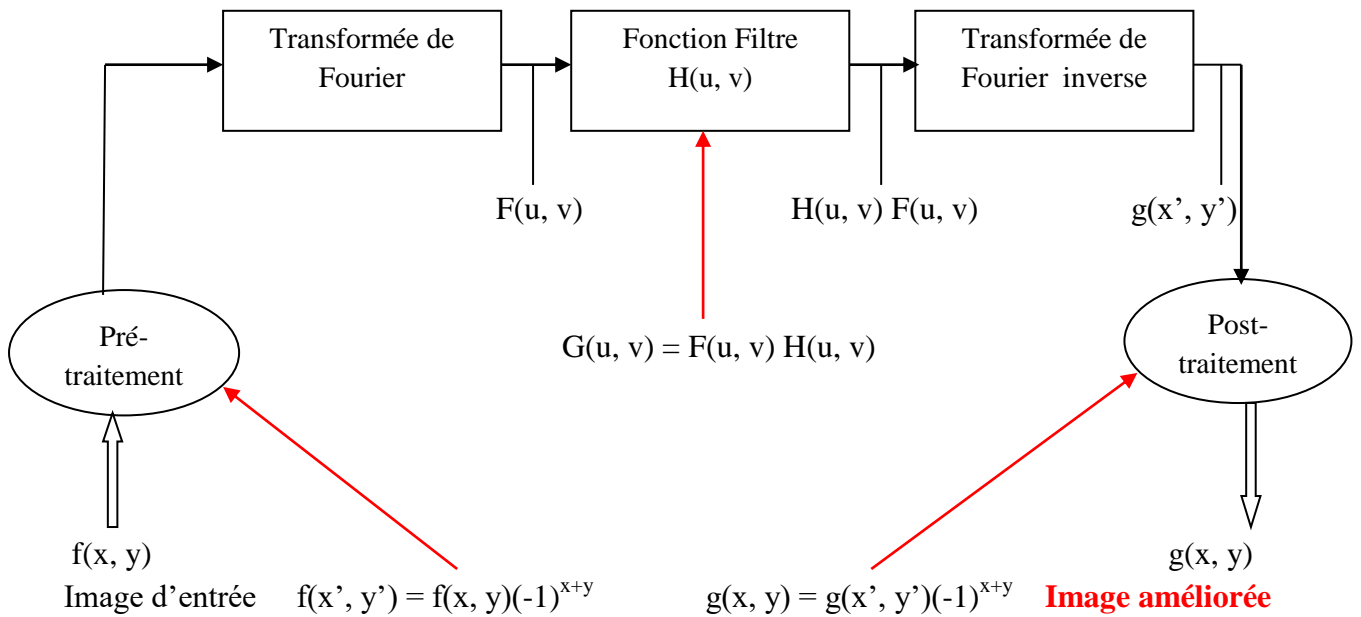
4- Calculer la transformée de Fourier inverse du résultat :

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

5- Extraire la partie réelle.

6- Multiplier cette partie réelle par  $(-1)^{x+y}$ .

Nous présentons sur la figure 2.38 un schéma résumant les étapes du filtrage dans le domaine fréquentiel.



**Figure 2.38 :** Filtrage dans le domaine fréquentiel.

Nous présentons sur la figure 2.39 l'application d'un filtre passe-bas sur une image dans le domaine de Fourier.

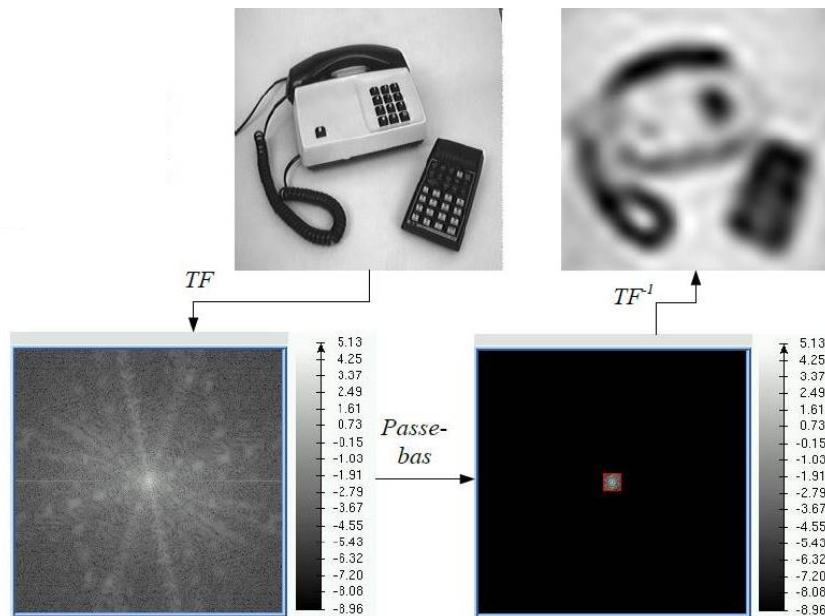


Figure 2.39 : Filtrage passe-bas dans le domaine de Fourier.

## Chapitre III : Éléments de théorie de l'information (entropie), MICD, compression de type JPEG/JPEG2000, psycho visuel.

### 1- L'entropie

La théorie de l'information permet d'évaluer la quantité d'information dans une image. Pour cela, chaque point d'une image est considéré comme une variable aléatoire. Une image est alors considérée comme une suite de  $W \cdot H$  variables aléatoires.

L'entropie est une grandeur qui caractérise la quantité d'informations que contient une image. Par exemple une image dont tous les pixels ont la même valeur contient très peu d'informations car elle est extrêmement redondante, son entropie est faible. En revanche, une image dont tous les pixels ont une valeur aléatoire contient beaucoup d'informations ; son entropie est forte.

En pratique, l'entropie d'une image numérique est inversement liée à la probabilité d'apparition des niveaux de gris dans l'image. Plus une valeur de gris  $k$  est rare, plus sa probabilité d'apparition  $p(k)$  est faible, et cela contribue à une entropie globale plus grande.

Par définition, l'entropie d'ordre zéro  $H_0$  est donnée par :

$$H_0 = - \sum_{k=0}^{2^R-1} p(k) \cdot \log_2 p(k) \quad \text{unité bpp}$$

Avec:  $p(k)$  est la probabilité d'apparition des niveaux de gris dans l'image,  $k$  est la valeur de gris et  $R$  est le nombre de bits par pixel.  $k$  désigne un point d'une image en niveaux de gris, c'est une variable aléatoire dont les valeurs sont des entiers de l'intervalle  $[0 \ 255]$  ( $2^{R-1}$ ) où 255 est la valeur maximale de niveau de gris. L'entropie est exprimée en bits par pixel et le logarithme est en base 2 car un bit peut avoir 2 valeurs différentes.

L'utilisation du logarithme de base 2, fait de  $H_0$  le nombre moyen de bits par pixel nécessaire pour coder toute l'information contenue dans l'image. Une image codée avec  $R$  bits par pixels a en fait presque toujours une entropie d'ordre zéro  $H_0$  inférieure à  $R$ .

Par conséquent, l'entropie  $H_0$  d'une image originale fournit le débit minimal qu'il est possible d'atteindre par compression pixel par pixel sans dégrader l'image, et par là même un taux de compression sans perte maximal.

Shannon définit également l'entropie d'un message, d'une manière comparable à la notion utilisée en thermodynamique. Dans cette discipline, il s'agit d'une grandeur physique qui permet d'évaluer la dégradation de l'énergie d'un système ; nous disons souvent que l'entropie mesure le degré de désordre d'un système. Dans la théorie de l'information, l'entropie d'un message indique le poids d'information qu'il porte, et donc sa "compressibilité" théorique.

### 2- Le codeur MICD

Le codeur MICD (en Anglais DPCM pour *Differential Pulse Code Modulation*) est un système simple, appartenant à la famille des codeurs prédictifs. Il en existe deux versions :

MICD : MIC Différentiel,

MICDA : MICD Adaptatif (celui auquel nous nous intéressons dépend du contenu des données image).

Les diagrammes en blocs du codeur et du décodeur sont présentés sur la figure 3.1, avec les notations suivantes :

$X(i, j)$  : signal d'entrée.

$X_p(i, j)$  : prédiction du signal d'entrée.

$E(i, j)$  : erreur de prédiction.

$\hat{E}(i, j)$  : erreur de prédiction quantifiée.

$\hat{X}(i, j)$  : signal codé-décodé (signal reconstruit qui sert à la prédiction de l'étape suivante).

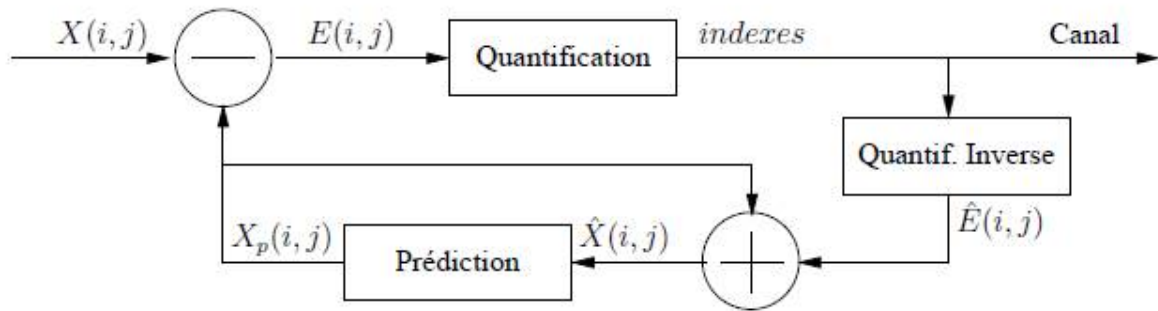
Un exemple que nous pouvons citer est le codage des signaux de télévision à traiter ou transmettre sous forme numérique est effectué en utilisant la modulation d'impulsions codées différentielles (MICD).

La motivation de ces codeurs appliqués en imagerie vient du fait que les pixels voisins (spatialement) ont une intensité lumineuse voisine (numériquement). Dans le schéma de codage, il s'agit de quantifier la différence  $E(i, j)$  entre le signal  $X(i, j)$  et une prédiction  $X_p(i, j)$  de ce signal. Cette prédiction est réalisée à partir des valeurs passées de  $\hat{E}$  et de  $X_p$ . Si nous notons  $Q$  le bruit de quantification, nous avons :

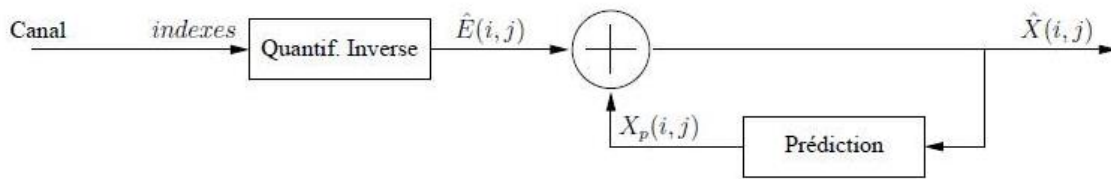
$$\hat{E}(i, j) = E(i, j) + Q(i, j)$$

En suivant la chaîne du codage, il vient :

$$\begin{aligned}\hat{X}(i, j) &= X_p(i, j) + \hat{E}(i, j) \\ &= X_p(i, j) + E(i, j) + Q(i, j) \\ &= X(i, j) + Q(i, j)\end{aligned}$$



(a) Schéma du codeur



(b) Schéma du décodeur

**Figure 3.1 :** Diagrammes du schéma de codage / décodage

Concernant la prédiction, Les pixels sont des valeurs d'intensité lumineuse et sont ici comprises entre 0 et 255. Pour les deux codeurs (MICD et MICDA), les pixels voisins du pixel à coder (noté X) sont notés comme suit :

Ligne précédente	B	C	D
Ligne courante	A	X	

Nous allons étudier deux prédictions fixes pour le MICD :

$P(X) = A$  (mono-dimensionnelle).

$P(X) = A + C / 2$  (bi-dimensionnelle).

### 3- Compression de type JPEG/JPEG2000, psycho visuelle

#### 3-1 Pourquoi vouloir compresser une image ?

De nos jours, la puissance des processeurs augmente plus vite que les capacités de stockage, et énormément plus vite que la bande passante des réseaux, car cela demande d'énormes changements dans les infrastructures de télécommunications. Ainsi, pour pallier ce manque, il est courant de réduire la taille des données en exploitant la puissance des processeurs plutôt qu'en augmentant les capacités de stockage et de transmission des données.

La compression d'images peut s'employer avec des contraintes et des attentes très différentes, selon l'utilisation à laquelle les images sont destinées. Nous pouvons vouloir réduire le nombre de bits d'une image avec une contrainte sur la capacité de stockage, la vitesse de transmission, le temps d'accès depuis un médium de stockage, etc... Nous pouvons prévoir d'utiliser l'image décompressée pour des applications de loisir, du traitement d'image, de la recherche visuelle rapide dans une base d'images, du diagnostic médical, etc...

Nous donnons quelques raisons de l'utilisation de la technologie de la numérisation en compression comme solution aux problèmes de stockage et de transmission de l'image :

- Une image au standard VGA+ (640 x 480, 256 couleurs) occupe un octet par pixel soit 300 ko. Dans cette définition, qui reste limitée par rapport aux standards audiovisuels, 25 images (soit une seconde de vidéo) occuperaient plus de 7 Mo, sans parler du son. Il est aisé de comprendre que ces valeurs sont incompatibles avec la plupart des supports informatiques actuels, et même avec les contraintes des réseaux publics ou privés envisageables à moyen terme.
- L'image consomme une quantité impressionnante d'octets quand elle est numérisée. Aujourd'hui, nous parlons de " qualité megapixel " pour les appareils photo numériques ; cela signifie que chaque image comporte environ un million de pixels dont chaque pixel nécessite trois octets pour les composantes RVB (rouge, vert, bleu). Donc, sans compression, cela représenterait un peu plus de 3 Mo pour une seule photographie.
- Une image de 800x800 pixels en 16 millions de couleurs (24 bits par pixel), correspondant à un fond d'écran, occupe 1 million 400 000 octets.
- La transmission d'une séquence d'images couleur au format QCIF échantillonnée à 30 Hertz représente un débit de 9.12 M bits par seconde, et de 36.40 M bits par seconde au format CIF (le format de QCIF [Quart de CIF]).
- La transmission d'une séquence vidéo 512x512, 8 bpp, 3 couleurs sur une ligne téléphonique avec modem à 9600 bauds (Le baud est une unité de mesure utilisée dans le domaine des télécommunications en général, et dans le domaine informatique en particulier, notamment dans le contexte de communications avec certains périphériques externes par exemple via un modem), nécessite 11 minutes par image.
- Une image de télévision basse résolution contient 3 composantes ou couleurs, 512x512 pixels par couleur et 8 bits par pixel, soit un total de  $6 \times 10^6$  bits.
- Les encyclopédies les plus complètes comportent environ 25 000 pages contenant du texte et des images. Nous pouvons considérer chaque page comme une image qui, numérisée à 300 dpi, occupe un volume de 1 Mo. Au total, la numérisation de l'ouvrage conduit à 25 Go. Il est clair que ce volume est impossible à stocker. Il faut donc effectuer une "compression" des données afin d'en réduire le volume.
- Dans d'autres domaines professionnels tels que l'imagerie médicale, des masses gigantesques de données sont acquises chaque jour. Nous chiffrons à environ 10 Teraoctets la masse de données d'un service radiologique d'un hôpital dans un pays industrialisé.

Pour remédier à ces contraintes, il n'y a qu'une solution : comprimer les images. Les chercheurs ont imaginé de nombreuses méthodes de compression, que nous les classons en deux catégories : celles qui se contentent de comprimer les données sans les altérer, et celles qui les compactent en les modifiant. Les premières, dites non destructives, permettent de reconstituer, au bit près, le fichier dans l'état où il était avant la compression.

Pour réduire le volume global des images tout en conservant l'image originale, un moyen consiste en la compression des images avec le minimum de dégradation et le maximum d'efficacité possible.

### **3.2 Définition de la compression d'image**

A cet effet, la compression et le codage consiste en la réduction de la taille physique d'un bloc d'information en réduisant le nombre de bits par pixel à stocker ou à transmettre suite à l'exploitation de la redondance informationnelle dans l'image. Nous parlons alors de réduction de redondance. La redondance peut être occasionnée par une représentation non efficace des données ; son élimination dans ce cas, n'empêche pas la restitution de ces données.

Trois sortes de redondances sont exploitées dans la compression d'images :

- La redondance spatiale entre pixels ou blocs voisins dans l'image ;
- La redondance temporelle entre images successives dans une séquence vidéo ;
- La redondance spectrale entre plan de couleur ou bandes spectrales.

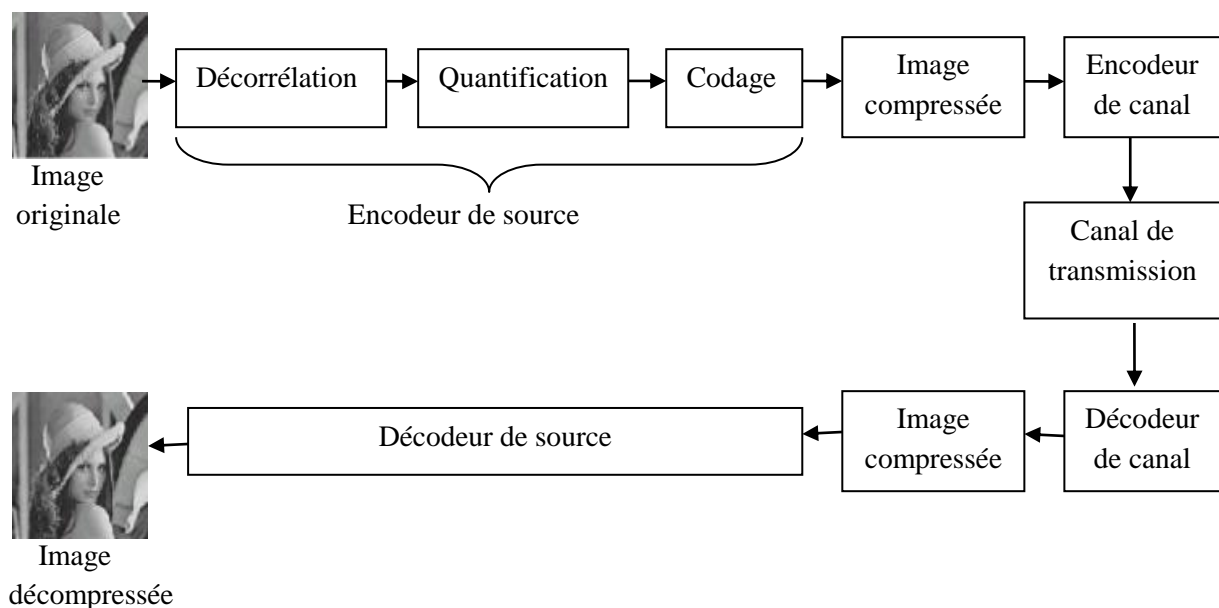
En fait, l'ensemble des techniques de compression des images exploite la corrélation entre des pixels voisins pour produire une " valeur probable " du prochain pixel à coder. Le produit de cette prédiction étant statistiquement proche de la valeur réelle, il est alors possible d'effectuer le codage de manière plus économique, en codant uniquement la différence avec la valeur prédite. Les ordres de grandeur des valeurs à transmettre sont alors statistiquement plus faibles, ce qui permet d'utiliser un nombre plus faible de bits.

### 3.3 La redondance psychovisuelle

L'œil humain perçoit une image de manière particulière ; il sait reconnaître des détails qui portent plus d'information que d'autres. Par suite, nous pouvons éliminer les détails qui ont très peu d'importance sans dégrader de manière significative la qualité de l'image ; cette opération est une quantification (opération irréversible). La codification IGS (Improved Gray Scale) est une méthode de codage permettant de prendre en compte les préoccupations développées ci-dessus. Supposons que l'on ait une image originale à 256 niveaux de gris. Nous pouvons réduire le volume de l'image en codant sur 16 niveaux ; malheureusement ces méthodes appliquées brutalement font apparaître des effets quelquefois inesthétiques ou purement artificiels (passages brusques d'un niveau de gris à un autre, faux contours notamment). La codification IGS est un moyen d'adoucir le procédé. Le principe est le suivant : le codage à 256 niveaux s'effectue sur 8 bits séparables en poids forts (4 bits) et poids faibles (4 bits) ; pour des pixels successifs, le codage IGS à 16 niveaux (donc sur 4 bits) s'obtient en tenant compte du codage sur 8 bits des pixels précédents et en ne retenant que les bits de poids forts.

### 3.4 Système de compression

La figure ci-dessous précise la chaîne d'opérations relatives à la compression/décompression d'images.



**Figure 3.2 :** Schéma d'un système de compression

Un système de compression comporte un encodeur à une extrémité du processus (source) et un décodeur à l'autre extrémité du processus (utilisation). Entre le codage de l'information source et le décodage pour utilisation existe un canal qui est la cause principale de l'opération de compression; par exemple, le canal pourra être un réseau de transmission de données.

### 3.4.1 L'encodeur de source :

L'objectif de l'encodeur 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. Nous avons mentionné auparavant que l'entropie d'une image désigne la quantité moyenne d'informations, le but de la réduction de l'entropie est par conséquent de transformer la distribution des valeurs de l'information à transmettre. L'encodeur de source (ou la réduction de l'entropie) est composé de trois étapes : la décorrélation, la quantification et le codage entropique.

Nous distinguons les étapes suivantes pour l'encodage de source :

- **Décorrélation.** la dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduit une corrélation très forte sur l'image. Nous essayons donc de tirer partie de cette corrélation, pour réduire le volume d'information en effectuant une opération de décorrélation des pixels ou réduction des redondances interpixels. La décorrélation consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés, c'est une opération réversible.

- **Quantification.** la quantification des coefficients a pour but de réduire le nombre de bits nécessaires pour leurs représentations. Elle représente une étape clé de la compression visant la réduction des redondances psychovisuelles.

- **Codage.** une fois les coefficients quantifiés, ils sont codés. Un codeur doit satisfaire à priori les conditions suivantes :

- Unicité : deux messages différents ne doivent pas être codés de la même façon,
- Déchiffrabilité : deux mots de codes successifs doivent être distingués sans ambiguïté.

Nous distinguons 2 types de codage de source:

1- Codage sans perte (« entropique »)

Ex: Huffman, Lempel-Ziv, Arithmétique, ...

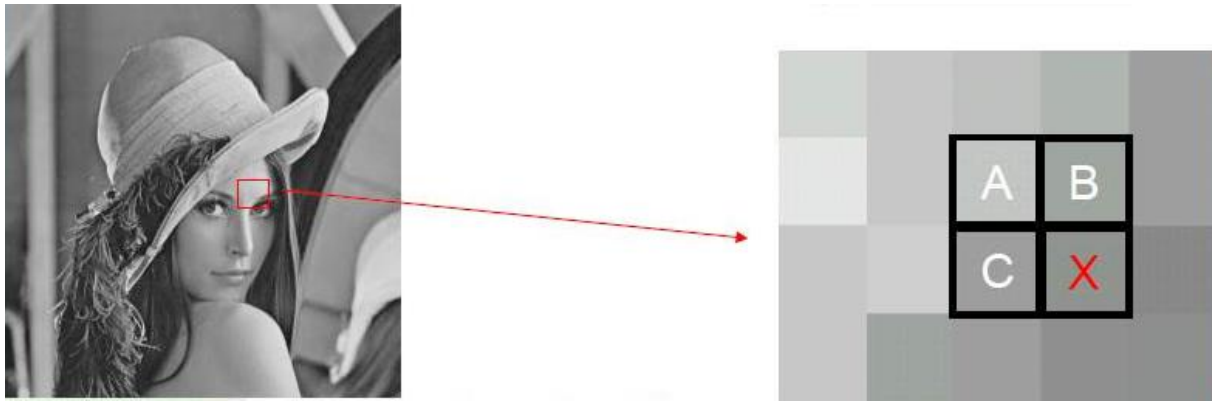
=> Taux de compression faible (1.5 à 2).

2- Codage avec pertes (« irréversible ») :

Suppression des redondances ou de l'information inutile car prévisible et invisible par le système visuel humain (SVH).

=> Taux de compression élevés (>10).

Ex: image transmise en la balayant (« Zigzag scan »).



**Figure 3.3 :** Principe du codage MICD

Nous utilisons le codage MICD (Modulation d'Impulsions Codées Différentielles): le niveau de gris d'un pixel dépend souvent de celui de ses voisins.

=> prédire X en utilisant A, B ou C.

Notons que de ces trois opérations, seule la quantification est une opération irréversible (puisque des données sont abandonnées).

### 3.4.2 L'encodeur de canal :

Le rôle du codage de canal est de protéger le message des perturbations du canal de transmission en ajoutant de la redondance au message compressé. En effet, l'encodeur de canal est nécessaire quand le canal est " bruiteux "; donc; les données sont codées dans un format permettant de réduire au maximum les erreurs dûes au passage de l'information dans le canal, c'est à dire d'accroître l'immunité au bruit. Pour les canaux du type réseau de transmission de données, les codes de Hamming sont les plus connus et les plus utilisés. Ils permettent de détecter les erreurs et de les auto-corriger.

### 3.4.3 Le décodeur de canal :

Nous le trouvons à l'autre extrémité de la chaîne, il effectue l'opération inverse de l'encodeur de canal.

### 3.4.4 Le décodeur de source :

Le décodeur de source se compose de deux étages :

- 1- Le décodage qui rétablit les redondances de codage.
- 2- La décorrélation inverse qui rétablit les redondances interpixel.

La quantification inverse évidemment n'a pas de sens, puisque l'opération directe est irréversible.

Les systèmes de compression utilisés réellement ne comportent pas nécessairement tous les blocs fonctionnels de la figure précédente. Nous pouvons d'ailleurs distinguer 2 catégories de systèmes de compression :

- Si  $I'(x,y) = I(x,y)$ , le système de compression est idéal ; il correspond à la compression sans pertes appelée quelquefois compactage.
- Si  $I'(x,y) <> I(x,y)$ , le système de compression admet une distorsion; il correspond à la compression avec pertes.

### 3.5 La compression JPEG

Les méthodes par transformation figurent parmi les techniques de compression les plus employées. Elles permettent d'obtenir des taux de compression élevés tout en conservant une bonne qualité d'image. Ce sont des méthodes qui font appel successivement à plusieurs principes de compression.

Pour le standard JPEG la base choisie est dérivée des bases de Fourier : l'image est découpée en blocs de taille 8x8 et chacun de ces sous-blocs est décomposé dans une base de cosinus locaux [Pennebaker & Mitchell, 1992 ; Gonzalez & Woods, 2018].

JPEG est un ensemble de spécifications et algorithmes de compression des images fixes. Leur statut de norme officielle constitue une évolution essentielle.

JPEG est originellement prévu pour des images en 16 millions de couleurs (24 bits par pixel). Tous les logiciels de traitement d'images exploitent actuellement le format JPEG, en importation comme en exportation. Il en va de même des appareils photos numériques. Des processeurs spécialisés sont également disponibles sur le marché.

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 suivante :

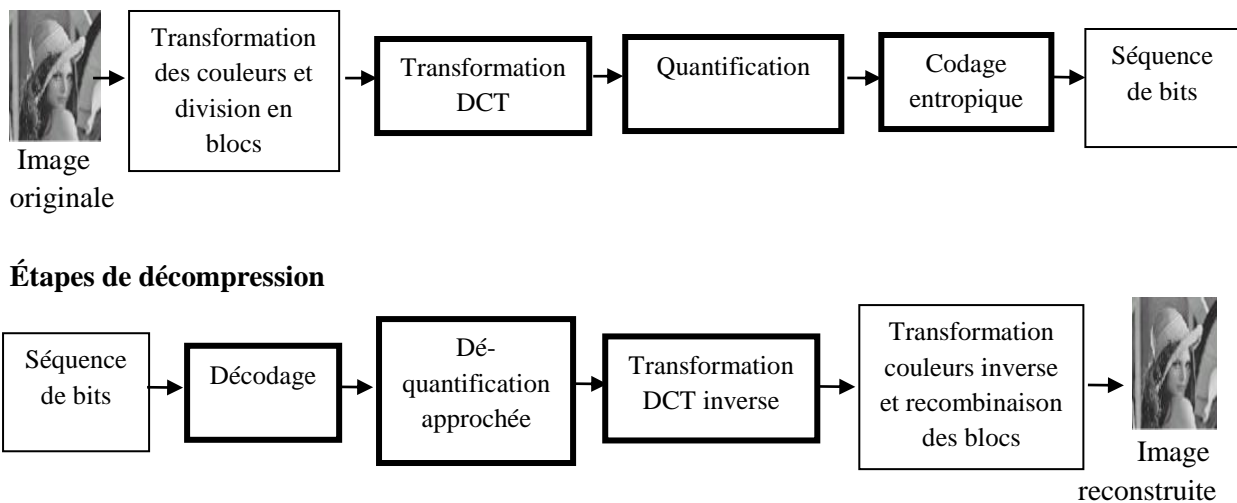


Figure 3.4 : Principe général de la compression/décompression JPEG.

#### 3.5.1 Transformation des couleurs et division en blocs

Le format JPEG, commence par traiter séparément chaque composante de couleur. Il est possible d'utiliser directement le modèle RVB, mais JPEG est plus efficace sur le modèle YCbCr (Y : la luminance, CbCr : la chrominance, utilisé pour stocker le traitement et transmettre la télévision et la vidéo numérique) après conversion éventuelle en luminance (à quel point le pixel est clair ou foncé) / chrominance (la couleur du pixel).

Pour constituer un pixel, nous aurons à définir une quantité de rouge et une de vert et une autre de bleu où nous pourrions déduire une des 3 valeurs en ayant les 2 autres. A cet effet, nous coderons uniquement le bleu et le rouge + la luminance (YCbCr) pour éviter d'alourdir l'information à transporter.

L'être humain est plus apte à détecter des variations de luminosité et moins apte à percevoir les différences de chrominance. En effet, durant la nuit nous conservons une bonne vision dès qu'il y a un peu de lumière tout en perdant la capacité à distinguer parfaitement les couleurs.

Le système YCbCr exploite le fait que le cerveau traduit le signal trichromatique perçu par l'œil, comme un signal composé de trois composantes, dont l'une est achromatique : la luminance. Elle permet d'éclaircir ou d'assombrir une couleur en ajustant la quantité de noir.

La méthode JPEG profite des imperfections de la perception d'une image par l'œil humain pour compresser sans dégradation apparente. Le système YCbCR facilite les traitements à effectuer sur une image fixe en tenant compte que l'œil humain :

– Est plus sensible à l'intensité (luminance) qu'à la couleur. On peut sous-échantillonner les composantes couleur (i.e. chrominance) avant leur compression.

– Perçoit mieux les contrastes sur les faibles que sur les fortes intensités, il est peu sensible aux variations en haute fréquence. On peut donc "quantifier" les variations de couleurs. (Quantifier un signal consiste à réduire sa précision en le discrétisant.)

La norme de JPEG tient compte du fait que le système visuel humain est moins sensible aux composantes chromatiques Cb et Cr qu'à la luminance, en sous-échantillonnant horizontalement et verticalement les composantes chromatiques avant leur compression. La luminance est prise en chaque pixel tandis que la chrominance est prise comme une valeur moyenne pour un bloc de pixels.

L'algorithme JPEG est basé sur cette différence de perception. D'ailleurs, il est même possible de sous échantillonner ces composantes de chrominance, afin d'aboutir à un modèle prédéfini. Ensuite le format JPEG, commence par découper chaque composante YCbCr en blocs carrés de 64 ( $8 \times 8$ ) pixels (L'algorithme de compression est calculé pour chaque bloc séparé).

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. Il peut résulter de cette division un effet visuel appelé effet de « blocs » : à des taux élevés la frontière des blocs devient visible car ils ont été comprimés indépendamment. La taille des blocs est variable selon les méthodes. Elle est de  $8 \times 8$  pour JPEG et MPEG.

### 3.5.2 Transformation DCT

La deuxième étape consiste à appliquer une transformation mathématique à chaque bloc. Les transformations utilisées en compression d'image sont des transformations orthogonales, leur but est de décorrélérer 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. La transformation fait passer d'un espace de nombres entiers, les pixels, à un espace de nombres flottants (voire de complexes) qui sont les coefficients du plan des fréquences, aussi appelés coefficients spectraux.

Les 2 motivations principales à l'utilisation d'une transformation sont :

- L'obtention d'une représentation de l'image qui se prête bien à la quantification et au codage.
- La possibilité d'ajuster les erreurs de quantification selon la sensibilité au système visuel humain.

Les algorithmes JPEG de compression non conservative sont fondés sur les propriétés de la DCT (« Discrete Cosine Transformation » ou « Transformation Discrète en Cosinus »).

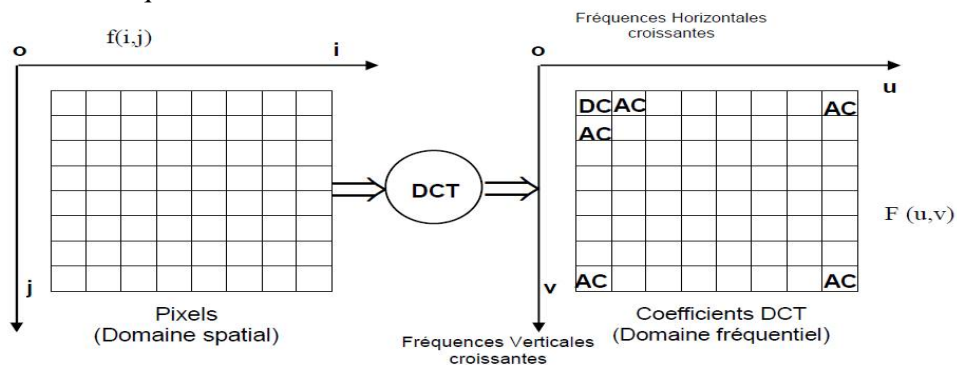
Les images couleur sont généralement codées comme trois images monochromes. L'image est tout d'abord découpée en zones de dimensions réduites (8 x 8 pixels), puis chaque pavé est traité par la DCT [Skodras et al., 2001]. La compression, selon un taux laissé au choix de l'utilisateur, est effectuée par quantification des coefficients issus de cette décomposition spectrale. Cette étape est la seule à l'origine des dégradations. Enfin, les données obtenues sont codées sous une forme elle-même compressée au moyen des algorithmes conservatifs (Huffman).

### 3.5.3 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'œil 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. En général, c'est le degré de quantification qui détermine le niveau de compression obtenu. Une quantification grossière donne une compression importante mais introduit une forte dégradation.

### 3.5.4 Illustration de l'algorithme JPEG avec perte

Nous présentons dans ce qui suit, les étapes de l'algorithme JPEG [Taubman & Marcellin, 2002] :  
 Étape1 : l'image est divisée en blocs de 8x8 pixels, une valeur optimale de taille de blocs pour JPEG est estimée à 8. Dans le codage JPEG, l'image est ainsi codée par blocs de 8x8 pixels. Nous effectuons alors la transformée en cosinus discrète (DCT) de chacun de ces blocs afin de passer du domaine spatial au domaine fréquentiel.



**Figure 3.5 :** Représentation de la DCT d'un bloc de 8 × 8 pixels.

La DCT permet schématiquement de changer l'échelle de mesure, en passant d'une échelle définissant un pixel en fonction de sa position en x et en y à une échelle définissant la fréquence d'apparition de ce pixel dans un bloc de pixels, en effet, il est dès lors possibles de supprimer des informations sans pour autant altérer le résultat final, contrairement à un bloc de pixels où la disparition brute de plusieurs éléments est immédiatement visible.

La formule de transformation DCT utilisée est :

$$X_c(k,l) = \frac{c(k,l)}{4} \sum_{m=0}^7 \sum_{n=0}^7 x(m,n) \cos \left[ \pi \frac{(2m+1)k}{16} \right] \cos \left[ \pi \frac{(2n+1)l}{16} \right]$$

Avec  $k=0, 1, \dots, 7$

$l=0, 1, \dots, 7$

Nous allons prendre le bloc suivant extrait à partir de l'image LENA :

$$x(m,n) = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix}$$

Dans ce qui suit, nous présentons la transformée du bloc, après avoir arrondi les coefficients fréquentiels à l'entier le plus proche :

$$X_c(k,l) = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

Nous voyons que la DCT a pour effet de concentrer l'information en très peu de coefficients fréquentiels correspondant aux basses fréquences, et que les autres coefficients sont de haute fréquence.

Dans la matrice, suite à l'application de la DCT, les basses fréquences se trouvent en haut à gauche et les hautes fréquences en bas à droite. Les hautes fréquences représentent les zones à forts contrastes dans l'image, i.e. changements rapides d'intensité des pixels. Une image classique admet une grande continuité entre les valeurs des pixels. Généralement les coefficients AC numériquement importants se trouvent dans le carré  $4 \times 4$  autour du coefficient DC. C'est le carré des "basses fréquences spatiales" (les valeurs gris ne montrent pas d'oscillations rapides). Ainsi on représente l'intégralité de l'information de l'image sur très peu de coefficients, correspondant à des fréquences plutôt basses.

Le coefficient 1260 sur la composante (0,0) représente "la moyenne" de la grandeur d'ensemble de la matrice d'entrée et ne dépend que de la moyenne des valeurs de gris. Il est appelé coefficient DC (direct current). Les 63 autres coefficients ne dépendent que des déviations de cette moyenne. Ils s'appellent coefficients AC (alternative current). Graphiquement la composante DC nous donne la valeur moyenne des niveaux du bloc, et les composantes AC représentent les amplitudes des fréquences spatiales horizontales et verticales dans le bloc.

Typiquement, les valeurs des pixels varient lentement de point au point adjacent à travers une image, en particulier pour des images qui ne présentent pas de discontinuités brutales. Par conséquent, les coefficients des basses fréquences, correspondant à des plages spatiales uniformes, ont des valeurs plus grandes que les coefficients des hautes fréquences. L'énergie du signal se trouve donc concentrée sur les fréquences spatiales inférieures, alors que la majorité des autres fréquences sont nulles ou presque.

Étape2 : les coefficients fréquentiels sont normalisés. Une matrice de normalisation composée d'entiers sur 8 bits est utilisée pour pondérer les coefficients. Chaque coefficient de la matrice transformée est divisé par la valeur correspondante de la matrice de normalisation. Les coefficients normalisés sont obtenus par la formule :

$$\hat{X}(k,l) = \text{int} \left[ \frac{X(k,l)}{q \cdot Q(k,l)} \right]$$

Où :

$\hat{X}(k,l)$  est le coefficient DCT normalisé.

$\text{Int}(X)$  est la fonction d'arrondi à l'entier le plus proche.

$X(k,l)$  est le coefficient DCT de position  $(k,l)$  du bloc.

$q$  est le pas de quantification.

$Q(k,l)$  est la matrice de normalisation.

La normalisation est équivalente à une quantification scalaire. Pour chaque composante (luminance, chrominance) la quantification consiste à diviser la matrice obtenue après avoir appliqué la DCT, par une autre, appelée matrice de quantification, et qui contient 8x8 coefficients. Cette division atténue les hautes fréquences auxquelles l'œil humain est très peu sensible. Les coefficients leur correspondant sont souvent ramenés à 0.

Quantifier un signal consiste à réduire sa précision en le discrétisant en quantités discontinues ou quanta. Un signal prenant valeur dans un alphabet de  $n$  valeurs sera représenté par un nombre inférieur  $m$  de valeurs. Le signal ainsi quantifié pourra être décrit avec moins de bits que la version initiale. Il existe donc, de fait, une erreur de quantification : nous introduisons des pertes, ce qui rend ainsi le processus irréversible.

Le pas de quantification est une valeur entière qui sert à régler le degré de quantification du bloc. Plus le pas est grand, plus grandes sont la compression ainsi que la distorsion. Dans les applications courantes du standard, les valeurs de  $q$  utilisées en pratique sont comprises entre 1 et environ 25. Le pas peut être modifié d'un bloc à l'autre afin d'adapter localement la compression.

La matrice de normalisation sert à pondérer les coefficients DCT afin d'adapter le degré de quantification de chaque coefficient en fonction de la sensibilité du système visuel au bruit de quantification. Ainsi, l'œil nu est peu sensible au bruit introduit dans les hautes fréquences et les valeurs de  $Q(k,l)$  correspondant aux fréquences élevées seront plus fortes que pour les basses fréquences. Prenons l'exemple d'une image couleur RGB+L, 4 matrices différentes de normalisation différentes peuvent être utilisées pour les composantes couleurs et luminance.

Une des matrices de normalisation de la luminance (respectivement quantification) fournie par le standard est :

$$Q(k,l) = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Sur l'exemple que nous avons utilisé, les coefficients normalisés sont :

$$\bar{X}_c(k,l) = \begin{bmatrix} 79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

La même matrice de normalisation est appliquée à tous les blocs de l'image. Elle est transmise avec les coefficients comprimés en tant que frais généraux de codage. Le choix de la matrice de normalisation est laissé libre à l'utilisateur. Le comité JPEG a effectué un travail considérable en vue de trouver expérimentalement des matrices qui fournissent les meilleurs résultats possibles visuellement en pondérant les coefficients selon leur importance perceptuelle.

Étape3 : le coefficient DC normalisé est codé.

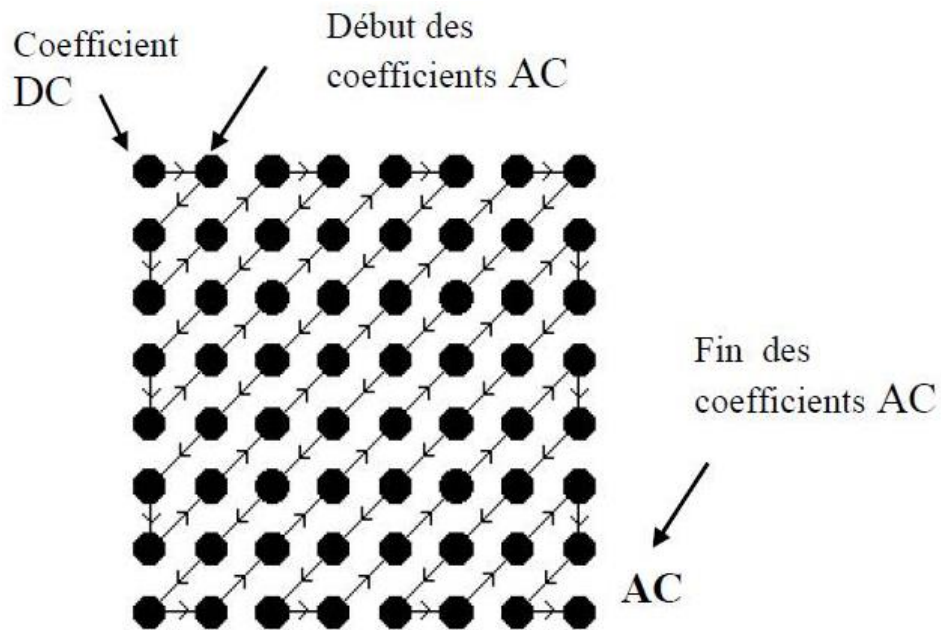
Par un système de DPCM, le coefficient DC est codé en utilisant la valeur DC du bloc précédent. Seul le coefficient du premier bloc d'image sera transmis sous sa forme absolue.

Étape4 : les coefficients AC sont réarrangés par une méthode zigzag.

L'énergie est concentrée dans les coefficients en haut à gauche de la matrice transformée. Nous réordonnons les coefficients afin d'obtenir un vecteur 1D qui range approximativement les coefficients par ordre décroissant d'énergie. Comme de nombreux coefficients de  $\bar{X}$  correspondant aux fréquences élevées sont nuls nous obtenons des grandes plages de zéros, qui se prêtent bien au codage RLC.

Au parcours ligne par ligne, nous préférons le Zig-zag scanning qui consiste à parcourir le bloc 8x8 de façon à lire en premier les coefficients forts puis de terminer par les plus faibles et d'augmenter la probabilité d'obtenir un nombre important de zéros consécutifs.

Le réarrangement zig-zag se fait comme mentionné sur la figure suivante :



**Figure 3.6 :** Réarrangement zig-zag des blocs normalisés JPEG

Dans l'exemple utilisé, le réarrangement zig-zag aboutit au vecteur suivant :

[79 0 -2 -1 -1 -1 0 0 -1 EOB]

Le symbole EOB est placé juste après la dernière valeur non nulle. Comme les blocs ont toujours 64 coefficients, il n'est pas nécessaire de préciser la longueur de la plage de zéro finissant le bloc réarrangé.

Étape5 : Le codage entropique

La suite des opérations est plus complexe : nous effectuons un codage particulier pour le coefficient le plus important (coefficient DC), puis un autre codage pour les autres coefficients (coefficients AC) :

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

a) table des catégories

catégorie	code de base	longueur
0	10	3
1	11	4
2	100	5
3	0	5
4	101	7
5	110	8
6	1110	10
7	11110	12
8	111110	14
9	1111110	16
A	11111110	18
B	111111110	20

b) table pour coefficients DC

1) Nous calculons la différence entre le coefficient DC : 79 et celui de la sous image précédemment traitée (par exemple 89) ; le résultat, -10, permet de déterminer, à partir de la table JPEG des catégories (a), la catégorie à prendre en compte (4) pour obtenir, dans la table JPEG de luminance (b), le code en binaire (101) et la longueur totale du code (7). Il faut donc compléter le code de base à 3 bits par 4 bits complémentaires ; ceux-ci sont générés par la règle suivante :

- Si la différence  $DC_i - DC_{i-1}$  est positive, on prend les K bits (où K est la catégorie précédemment déterminée) les moins significatifs de  $DC_i$ .
- Si la différence  $DC_i - DC_{i-1}$  est négative, on prend le K bits les moins significatifs de  $DC_i$  et on retranche 1. Nous sommes ici dans le second cas :  $-11 = (0100)_2$ . On a donc finalement pour le code de DC : 1010100.

2- Pour le codage des coefficients AC, on utilise les tables données en (a), (b) et (c). On notera sur (c) que le nombre de " 0 " devant un coefficient AC non nul est pris en considération. On peut ainsi, pour chaque coefficient AC, déterminer sa catégorie, puis son code de base et sa longueur ; les bits complémentaires sont obtenus avec la règle suivante : si le coefficient est positif, on prend les bits les moins significatifs ; si le coefficient AC est négatif, on retranche 1 et on prend les bits les moins significatifs.

Il est utile de mentionner quelques remarques :

Un coefficient non zéro dans la lecture séquentielle d'un schéma quantifié est caractérisé par :

1. Le nombre des zéros qui le séparent de son prédécesseur non-nul : 0-tête.
2. Sa catégorie i qui donne le nombre de bits nécessaires pour le codage du coefficient.
3. Son numéro dans la catégorie (les numéros commencent par zéro).

Pour effectuer le codage de Huffman de la lecture séquentielle des coefficients quantifiés, il nous manque seulement une table pour les mots code des symboles du type 0-tête/catégorie.

Parce que les zéros peuvent être de chaînes de plus en plus longues, le nombre 0-tête est considéré un nombre binaire en base hexadécimale : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f.

Si le nombre 0-tête est plus grand que 16 on rajoute un symbole zero run list pour chaque bloc de 16 zéros et on compte la suite de zéros.

4. Il y a donc deux symboles particuliers à coder :

(EOB)  $\equiv$  end of block qui indique la fin des coefficients non-nuls dans la suite des 63 coefficients à coder.

2. (ZRL)  $\equiv$  zero run list qui code un bloc de 16 entiers 0.

La table pour le codage des valeurs luminance (transformées et quantifiées) a été établie empiriquement par le groupe JPEG (Joint Photographic Experts Group), à la base d'un grand nombre de tests avec des images à précision 8 bits. Elle n'est pas vraiment une table par défaut, mais communément acceptée en tant que telle.

Représentation en binaire d'un nombre négatif est une représentation en complément à 2 :

– Partir de la représentation positive de ce nombre.

– Inverser les valeurs de tous les bits : les 1 deviennent des 0 et les 0 des 1.

– Ajouter 1 au résultat de l'opération précédente, ou bien rajouter un bit de "signe" avant.

En utilisant les catégories faites selon le nombre minimal de bits nécessaires au codage nous n'avons plus besoin d'utiliser le bit supplémentaire. Pourquoi ?

Le code du n°17 dans la catégorie 6, i.e. codage sur 6 bits est : le complément à 2 de 101110 : 010001

Le code du n°3 dans la catégorie 3, i.e. codage sur 3 bits est : le complément à 2 de 100 : 011

Le code du n°0 dans la catégorie 1, i.e. codage sur 1 bit est : le complément à 2 de 1 : 0

c) Table pour les coefficients AC

0 – tête/cat.	mot code	0 – tête/cat.	mot code
0/0( <i>EOB</i> )	1010	3/9	111111110010100
0/1	00	3/ <i>a</i>	111111110010101
0/2	01	4/1	111011
0/3	100	4/2	111111000
0/4	1011	4/3	111111110010110
0/5	11010	4/4	111111110010111
0/6	1111000	4/5	111111110011000
0/7	11111000	4/6	111111110011001
0/8	1111110110	4/7	111111110011010
0/9	111111110000010	4/8	111111110011011
0/ <i>a</i>	111111110000011	4/9	111111110011100
1/1	1100	4/ <i>a</i>	111111110011101
1/2	11011	5/1	1111010
1/3	1111001	5/2	11111110111
1/4	111110110	5/3	111111110011110
1/5	11111110110	5/4	111111110011111
1/6	111111110000100	5/5	1111111110100000
1/7	111111110000101	5/6	1111111110100001
1/8	111111110000110	5/7	1111111110100010
1/9	111111110000111	5/8	1111111110100011
1/ <i>a</i>	111111110001000	5/9	1111111110100100

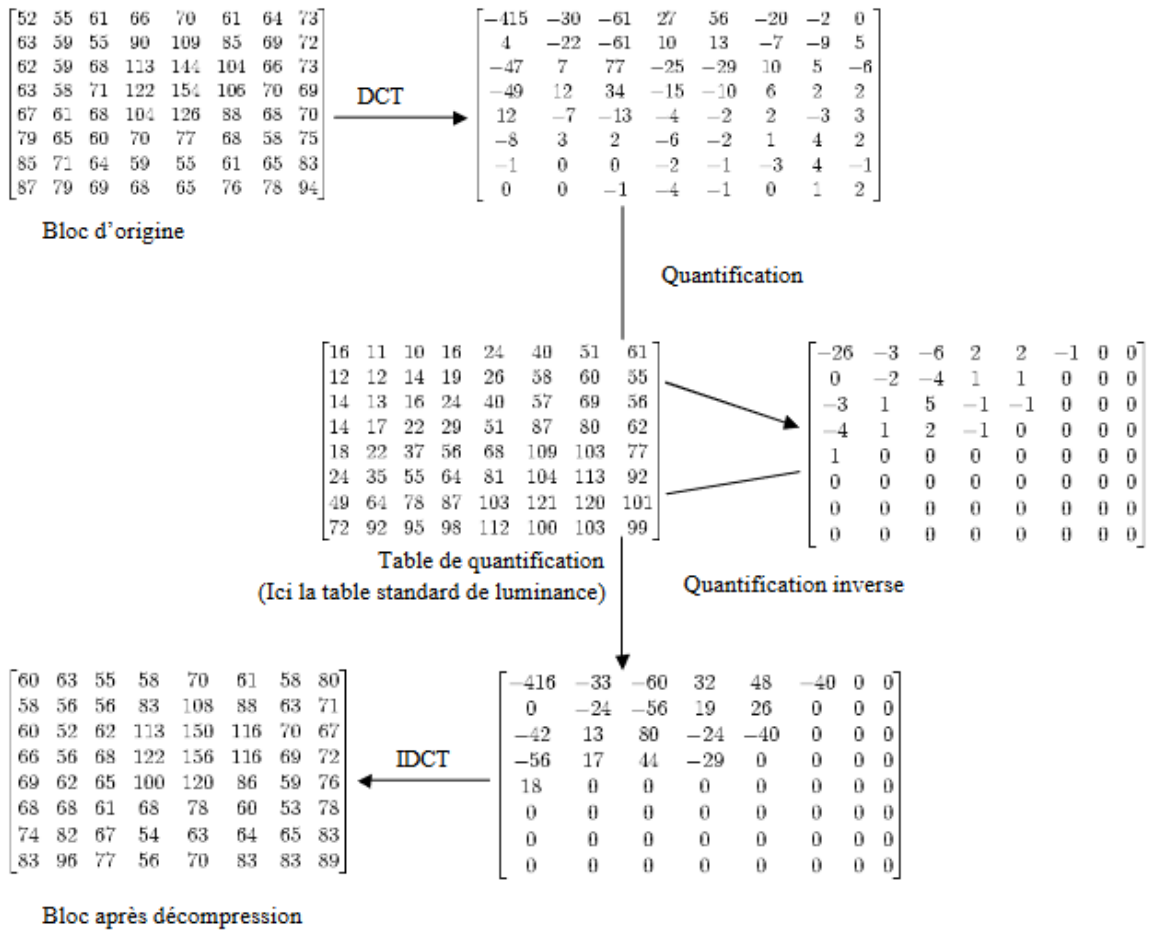
2/1	11100	5/a	111111110100101
2/2	11111001	6/1	1111011
2/3	1111110111	6/2	111111110110
2/4	111111110100	6/3	1111111110100110
2/5	1111111110001001	6/4	1111111110100111
2/6	1111111110001010	6/5	1111111110101000
2/7	1111111110001011	6/6	1111111110101001
2/8	1111111110001100	6/7	1111111110101010
2/9	1111111110001101	6/8	1111111110101011
2/a	1111111110001110	6/9	1111111110101100
3/1	111010	6/a	1111111110101101
3/2	111110111	7/1	11111010
3/3	111111110101	7/2	111111110111
3/4	1111111110001111	7/3	1111111110101110
3/5	1111111110010000	7/4	1111111110101111
3/6	1111111110010001	7/5	1111111110110000
3/7	1111111110010010	7/6	1111111110110001
3/8	1111111110010011	7/7	1111111110110010

7/8	1111111110110011	b/7	1111111111010101
7/9	1111111110110100	b/8	1111111111010110
7/a	1111111110110101	b/9	1111111111010111
8/1	111111000	b/a	1111111111011000
8/2	111111111000000	c/1	1111111010
8/3	1111111110110110	c/2	1111111111011001
8/4	1111111110110111	c/3	1111111111011010
8/5	1111111110111000	c/4	1111111111011011
8/6	1111111110111001	c/5	1111111111011100
8/7	1111111110111010	c/6	1111111111011101
8/8	1111111110111011	c/7	1111111111011110
8/9	1111111110111100	c/8	1111111111011111
8/a	1111111110111101	c/9	1111111111100000
9/1	111111001	c/a	1111111111100001
9/2	1111111110111110	d/1	11111111000
9/3	1111111110111111	d/2	1111111111100010
9/4	1111111111000000	d/3	1111111111100011
9/5	1111111111000001	d/4	1111111111100100

9/6	1111111111000010	<i>d/5</i>	1111111111100101
9/7	1111111111000011	<i>d/6</i>	1111111111100110
9/8	1111111111000100	<i>d/7</i>	1111111111100111
9/9	1111111111000101	<i>d/8</i>	1111111111101000
9/ <i>a</i>	1111111111000110	<i>d/9</i>	1111111111101001
<i>a/1</i>	111111010	<i>d/a</i>	1111111111101010
<i>a/2</i>	1111111111000111	<i>e/1</i>	1111111111101011
<i>a/3</i>	1111111111001000	<i>e/2</i>	1111111111101100
<i>a/4</i>	1111111111001001	<i>e/3</i>	1111111111101101
<i>a/5</i>	1111111111001010	<i>e/4</i>	1111111111101110
<i>a/6</i>	1111111111001011	<i>e/5</i>	1111111111101111
<i>a/7</i>	1111111111001100	<i>e/6</i>	1111111111110000
<i>a/8</i>	1111111111001101	<i>e/7</i>	1111111111110001
<i>a/9</i>	1111111111001110	<i>e/8</i>	1111111111110010
<i>a/a</i>	1111111111001111	<i>e/9</i>	1111111111110011
<i>b/1</i>	1111111001	<i>e/a</i>	1111111111110100
<i>b/2</i>	1111111111010000	<i>f/0(ZRL)</i>	11111111001
<i>b/3</i>	1111111111010001	<i>f/1</i>	1111111111110101
<i>b/4</i>	1111111111010010	<i>f/2</i>	1111111111110110
<i>b/5</i>	1111111111010011	<i>f/3</i>	1111111111110111
<i>b/6</i>	1111111111010100	<i>f/4</i>	1111111111111000
<i>f/5</i>	1111111111111001	<i>f/8</i>	1111111111111100
<i>f/6</i>	1111111111111010	<i>f/9</i>	1111111111111101
<i>f/7</i>	1111111111111011	<i>f/a</i>	1111111111111110

Exemple : Considérons un exemple de luminance d'un dégradé, son schéma transformé, puis le schéma quantifié. La quantification s'effectue en faisant opérer un schéma fixe de 8 fois 8 diviseurs, suivi d'un arrondi à l'entier le plus proche.





## Préambule

Claude Shannon, ingénieur chercheur en télécommunication, est l'un des principaux initiateurs de la théorie de l'information. Cette théorie a pour objectif de mesurer la quantité d'informations émises par un message. Mais qu'est-ce qu'une information ?

En termes de télécommunication, un message qui n'est ni un bruit ni un signal purement aléatoire, contient une information. Cette information peut être perturbée par des redondances ou bien par des effets extérieurs indésirables, appelés bruits, comme la perte ou l'interférence du signal.

Dans ce contexte, mesurer la quantité d'informations revient alors à mesurer le nombre minimal de bits nécessaires au codage et à la quantification d'un signal sans en déformer l'information ; il s'agit donc d'identifier et d'éliminer les redondances et le bruit du signal.

Ces travaux sont à l'origine de la théorie de l'information dont l'un des outils majeurs est l'entropie. Cette théorie est la base des deux processus de la quantification et le codage entropique.

Shannon dans sa théorie mathématique de la communication a défini la notion d'information en affirmant que tout message peut se définir sous forme de suite de 0 et 1. D'autres part, les signaux porteurs d'information : audio, image, vidéo... sont tous de nature analogique, ceci dit, les ordinateurs sont des dispositifs qui traitent des données représentées par une suite de nombre. En effet, une incompatibilité est distinguée.

A cet effet, pour relier ces deux univers il était nécessaire de fabriquer des composants transformant ces signaux initiaux continus en grandeurs manipulables par les processeurs.

De multiples applications de la technologie moderne utilisent des sons et des images numériques, voir les CD, DVD, appareils photos, caméscopes, téléphone, radio, les ordinateurs, internet ... d'où l'intérêt de bien cerner les tenants et les aboutissants de l'opération de numérisation. Dans ce sens, l'objectif principal de ce cours est la caractérisation d'un signal dans le domaine temporel et fréquentiel pour aboutir à des modèles mathématiques. La description mathématique des signaux permet de concevoir et de caractériser des systèmes de traitement de l'information.

L'ultime but est de comprendre :

- Le choix de  $T_e$ , période d'échantillonnage.
- Le Choix de  $n$ , nombre de bit de code.

- L'influence de l'échantillonnage sur les propriétés d'un signal.

Nous devons garder à l'esprit le fait que l'acquisition numérique ne doit pas détériorer le signal. On doit conserver au travers de la numérisation l'information utile :

Voix : [0 ;20kHz].

De plus, il faut limiter l'espace mémoire nécessaire au stockage. En effet, il faut stocker «  $n \cdot F_e$  » bits par seconde. On s'attachera dans une chaîne d'acquisition à minimiser cette valeur tout en ne détériorant pas le signal.

## 1- Définition d'un signal

Un **signal** est un message simplifié et généralement codé. Il existe sous forme d'objets ayant des formes particulières.

- Les **signaux lumineux** sont employés depuis la nuit des temps par les hommes pour communiquer entre eux à distance.
- Le **signal électrique** est une des formes les plus récentes de signal.
- Un **signal** dans le domaine informatique et de la communication interprocessus.

On a l'habitude de représenter un signal par une fonction continue dans le temps et de visualiser le signal sur un oscilloscope ou un appareil représentant la variation d'amplitude d'un phénomène en fonction du temps (cardiogramme, sismographe, microphone, ...).

## 2- Définition d'un Son

Le son est une vibration de l'air qui se propage avec des caractéristiques variables d'intensité, de fréquence, de portée, d'écho, ...

L'oreille humaine est sensible aux sons dans certaines limites d'intensité et de fréquence, c'est le processus de l'audition.

Quand les cordes vocales créent des sons, c'est la voix et le processus de la phonation.

Pour simplifier l'étude nous allons prendre l'exemple d'une seule corde (remarque : La production des sons à partir des cordes a été étudiée depuis Pythagore : en pinçant une corde, la vibration de la corde est transmise à l'air environnant, et l'air transmet la vibration de proche en proche jusqu'à votre oreille qui reçoit la vibration du début un peu transformée, mais encore reconnaissable. Tous ces phénomènes de transmission de "proche en proche" sont dus aux phénomènes d'onde :

Avant (il y a une bosse) :

Après (la bosse est plus loin) :



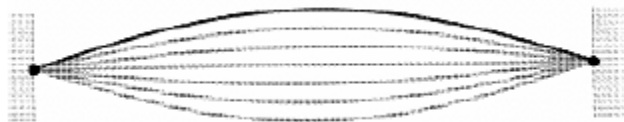
**Figure 1** : Evolution de l'onde en fonction de la distance

Exemple de la corde d'une guitare.

Si on fait vibrer une corde suffisamment tendue (souvent en nylon ou en acier), elle fait aussi vibrer l'air qui se trouve à côté et produit un son. Mais ce son est très faible.

C'est pour ça que les instruments à cordes ont des résonateurs (table d'harmonie).

La corde d'une guitare est attachée aux deux extrémités, lorsqu'on écarte la corde à vide, la déformation de la corde forme une onde entre les deux extrémités (un fuseau).



**Figure 2** : Représentation d'un fuseau, variation de l'amplitude de l'onde sur une corde fixée aux deux extrémités

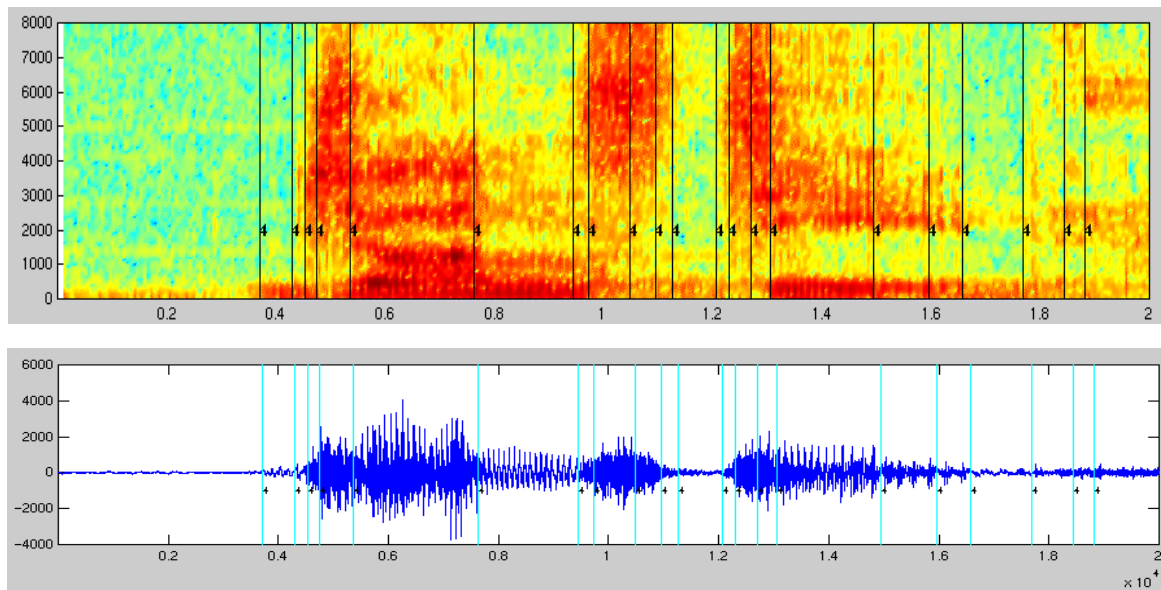
### 3- Définition d'un document sonore

Un document sonore est constitué de nombreuses composantes ou sources, dont les plus communes sont la parole et la musique. Evidemment, ces deux principales sources de sons ne sont que des composantes primaires, auxquelles il convient de rajouter les multiples sources de bruit qui existent.

Avant d'aller plus loin, il convient de préciser certaines définitions, notamment sur la parole et la musique.

#### 3.1 Parole

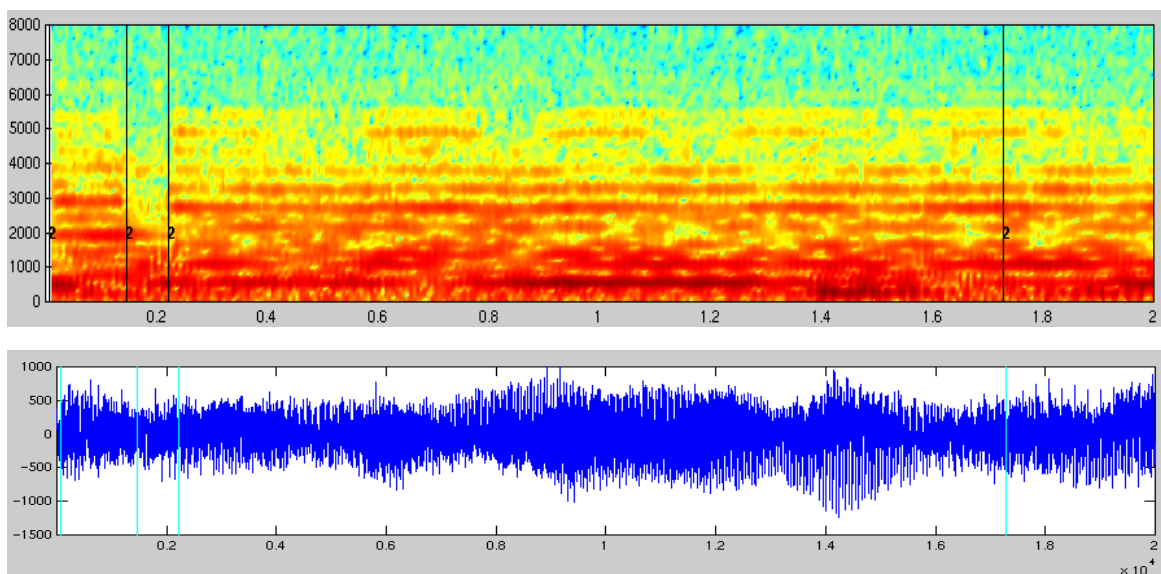
La parole est une suite de sons produits soit par **des vibrations des cordes vocales** (source quasi périodique de voisement), soit par **une turbulence créée par l'air s'écoulant dans le conduit vocal ou lors du relâchement d'une occlusion de ce conduit** (sources de bruit non voisées). Les sons voisés possèdent une structure formantique. La durée d'un son est de l'ordre de 60 à 100ms (cf. spectrogramme de la figure 3).



**Figure 3 :** Spectrogram de parole : alternance de sons voisés et non voisés sur 1,2s. Dans le cas voisé, une structure formantique est présente.

### 3.2 Musique

Définir la musique est très difficile car celle-ci peut être produite de différentes manières. C'est pourquoi, de nombreux chercheurs se limitent (lorsqu'il s'agit d'extraire cette composante) à de la musique « instrumentale traditionnelle » dans le sens où elle est une composition de sons harmoniques (de notes au sens classique), dans un cadre éventuellement polyphonique ; il s'agit alors de trouver l'existence d'une (des) hauteur(s) ou fréquence(s) fondamentale(s) (cf. spectrogramme de la figure 4).

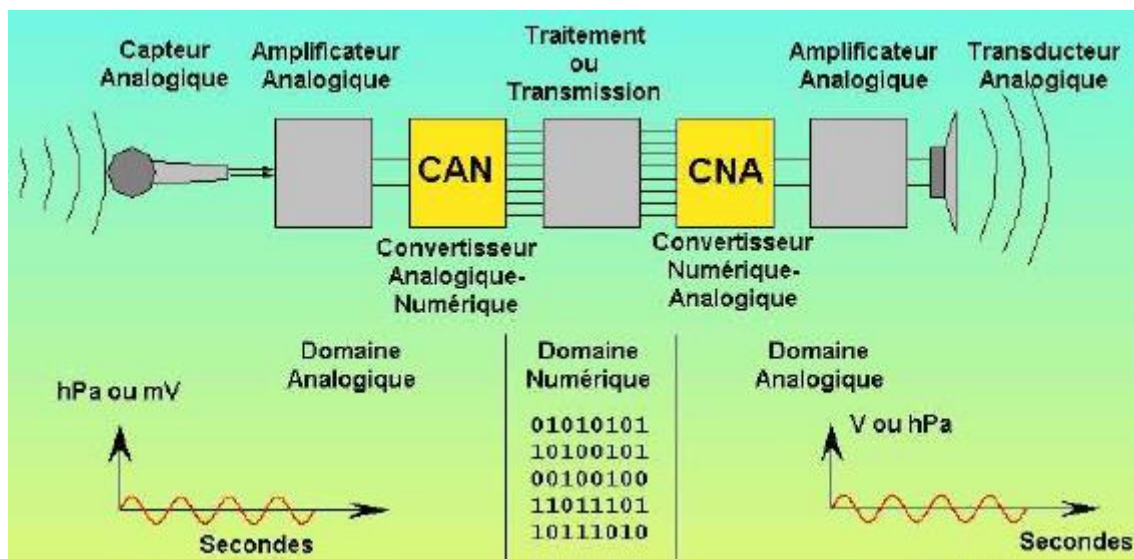


**Figure 4 :** Spectrogramme de musique traditionnelle : structure harmonique et présence d'une fréquence fondamentale ou hauteur de durée 1,2s.

#### 4- Numérisation d'un signal audio

Un son désigne une vibration que nous pouvons visualiser le signal électrique associé grâce à un micro relié à un portable ou à un ordinateur.

Nous distinguons les étapes suivantes (comme mentionné sur la figure 5) pour la numérisation d'un signal audio :



**Figure 5 :** Les étapes de numérisation d'un signal audio

- 1- Transformer le signal analogique en signal électrique : capture/acquisition du signal. Un microphone est relié à la carte son d'un ordinateur (la tension électrique à la borne du microphone que l'on peut visualiser à l'aide d'un oscilloscope nous montre une variation continue de cette tension : une reproduction très fidèle du signal sonore)
- 2- Convertir le signal électrique en une suite de valeurs numériques binaires : convertisseur analogique/numérique CAN (carte son).
- 3- Lancer le programme de calcul mathématique censé opérer le traitement voulu.
- 4- Convertir les codes binaires résultant du calcul en un signal électrique : convertisseur numérique/analogique CNA.
- 5- Reconvertir le signal électrique dans la grandeur physique initiale : haut-parleurs représentant des transducteurs électroacoustiques capables de traduire un signal électrique en vibration sonore de l'air, d'où un son. C'est l'inverse du micro.

Ce procédé de numérisation d'un signal audio consiste en pratique à le découper doublement :

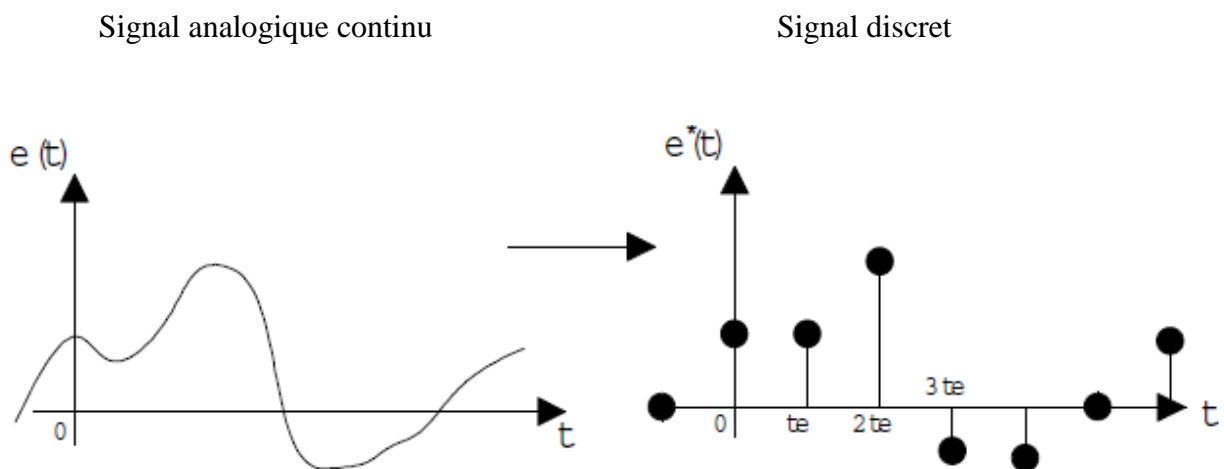
- 1- D'abord selon l'axe des temps, et on nomme *échantillonnage* ce découpage et *période d'échantillonnage* l'épaisseur des tranches de temps
- 2- Ensuite, selon l'axe des intensités, et on nomme *quantification* ce découpage dont la finesse des tranches est le *pas de quantification*.

## 4.1 Théorie de l'échantillonnage

### 4.1.1 Acquisition des Signaux

Pour transformer un signal analogique en un signal numérique, il faut le discrétiser.

On va donc prélever régulièrement des échantillons du signal analogique pour le rendre discret et permettre ainsi sa numérisation comme mentionné sur la figure 6 :



**Figure 6 :** Allure d'un signal échantillonné

On prend ainsi des valeurs de  $e(t)$  à des intervalles de temps régulier (tous les  $Te$ , période d'échantillonnage) à une fréquence  $Fe$  dite fréquence d'échantillonnage, que l'on déterminera par la suite. Suite à cet échantillonnage, on quantifie chaque échantillon par une valeur binaire pour la stocker sur un support numérique.

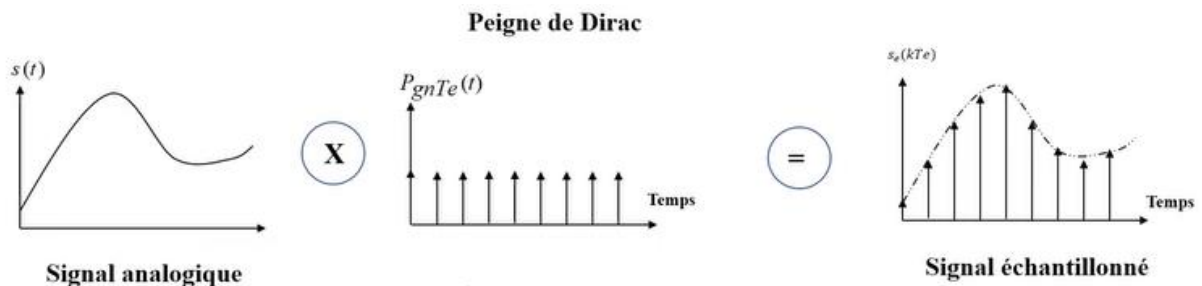
### 4.1.2 Modélisation de l'échantillonnage

L'opération mathématique associée à cette discrétisation revient à multiplier le signal  $e(t)$  par un peigne de Dirac  $\delta_{T_e}(t)$  :

$$e^*(t) = e(t) \cdot \delta_{T_e}(t) = e(t) \cdot \sum \delta(t - nT_e)$$

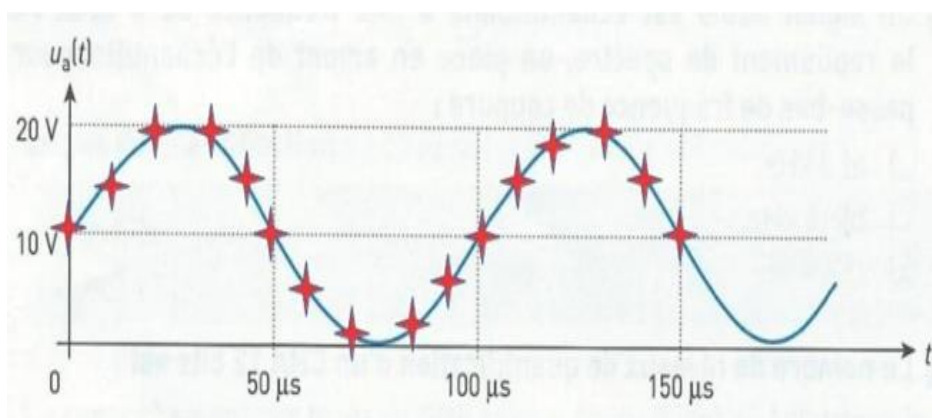
On peut ainsi calculer la transformée de Fourier du signal échantillonné en utilisant les propriétés liant une multiplication temporelle qui dans l'espace fréquentiel devient un produit de convolution. Donc, le signal échantillonné dans le domaine fréquentiel n'est autre que le produit de convolution entre le signal et le peigne de Dirac.

Echantillonner le signal  $e(t)$  dans le domaine temporel, revient donc à recopier dans le domaine fréquentiel son spectre  $E(f)$  tous les  $F_e$  comme mentionné sur la figure suivante :



**Figure 7 :** Les étapes du processus d'échantillonnage

Prenons l'exemple d'un signal sinusoïdal de la figure 8 :



**Figure 8 :** Signal sinusoïdal

La fréquence de ce signal est de : 10 khz

La fréquence d'échantillonnage est de : 100 khz

Ceci veut dire que nous prendrons des mesures à partir du signal tous les  $T_e$  secondes (période d'échantillonnage).

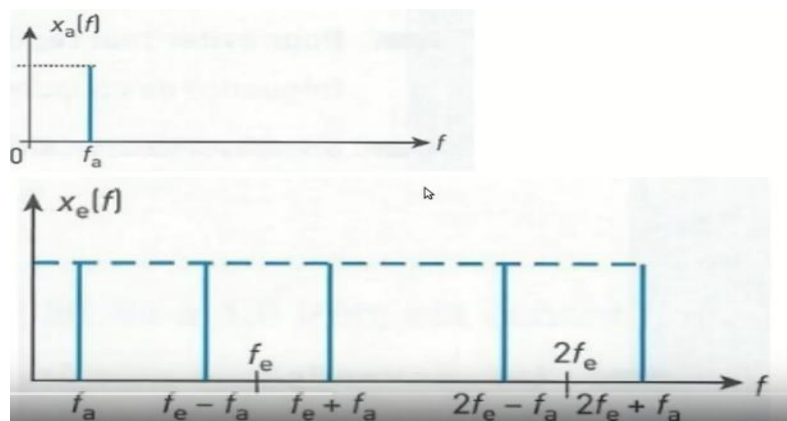
$$T_e = 1/f_e = 1/10^2 \times 10^3 \text{ (pour convertir au hz)} = 1/10^5 = 10^{-5} \text{ s}$$

Nous devons convertir la période de l'échantillonnage en micro seconde comme suit :

$$1 \mu\text{s} = 10^{-6} \text{ s} \rightarrow 1 \text{ s} = 10^6 \mu\text{s} \rightarrow 10^{-5} \times 10^6 \mu\text{s} = 10 \mu\text{s}$$

Les échantillons sont prélevés chaque les  $10 \mu\text{s}$  (nous pouvons bien le constater sur la figure ci-dessus).

Nous présentons sur la figure suivante le spectre du signal ainsi que du signal échantillonné :



**Figure 9 :** Spectre du signal sinusoïdal ainsi que du signal sinusoïdal échantillonné

Nous pouvons affirmer que le spectre du signal sinusoïdal contient une seule raie placée à la fréquence du signal ( $f_a = 10 \text{ kHz}$ ). Tandis que le signal échantillonné présente un spectre plus complexe (un ensemble de points au lieu d'une sinusoïde lisse). En effet, nous observons la raie de la fréquence du signal ainsi que des raies supplémentaires générées par l'échantillonnage (autours de la fréquence d'échantillonnage  $f_e$  (100 kHz) et de  $2 f_e$  à savoir : 90 kHz ( $f_e - f_a$ ) 110 kHz ( $f_e + f_a$ ) 190 kHz ( $2f_e - f_a$ ) 210 kHz ( $2f_e + f_a$ ) ...).

Nous allons considérer une nouvelle fréquence d'échantillonnage comme suit :

La fréquence de ce signal est de : 10 kHz

La fréquence d'échantillonnage est de : 13 kHz

Dans ce cas-là les raies générées par l'échantillonnage seront :  $f_e - f_a = 3 \text{ kHz}$ ,  $f_e + f_a = 23 \text{ kHz}$ ,  $2f_e - f_a = 16 \text{ kHz}$ ,  $2f_e + f_a = 36 \text{ kHz}$

Nous observons l'apparition de la raie de **3kHz** qui est inférieure à la raie de la fréquence du signal, et comme conséquence une fois nous désirerons revenir au signal initial après le traitement numérique nous aurons à enlever toutes les raies générées suite à l'échantillonnage (inexistantes dans le monde analogique) et ceci par l'application d'un filtre enlevant les raies

de fréquence supérieur à celle du signal mais pas celles inférieures et à cet effet le signal original changera d'allure !

Nous appelons ce phénomène par **le repliement de spectre** (la non possibilité de la restitution du signal analogique), il n'est plus possible de retrouver le spectre du signal d'origine. Dans ce cas, l'opération d'échantillonnage modifie les caractéristiques du signal d'entrée.

Par l'application d'un filtre passe-bas de fréquence de coupure proche à celle du signal, ceci ne règlera pas le problème car le filtre enlèvera les raies de fréquences supérieures à celle de la fréquence du signal et non pas celles de fréquence inférieure à **3khz**.

Donc, l'utilisation d'un filtre passe-bas (dans le domaine fréquentiel évidemment) est idéale pour la restitution du signal de départ mais à condition que y'aura pas de repliement de spectre.

#### 4.1.3 Théorème de Shannon

Nous nous trouvons devant l'obligation de choisir une fréquence d'échantillonnage suffisamment grande pour que la première raie soit supérieure à la fréquence du signal comme suit :

$$f_e - f_a > f_a$$

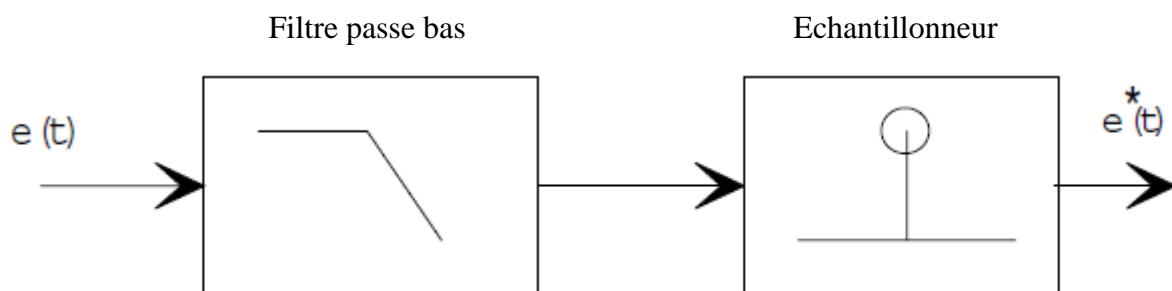
Ainsi, si l'on ne veut pas perdre d'informations par rapport au signal que l'on échantillonne, on devra toujours respecter la condition : ( $f_e \geq 2f_{max}$ ). Condition plus connue par le théorème de Shannon. En effet, sur la figure 7 il suffit que :

$f_e - f_a > f_a \rightarrow f_e > 2f_a$  (la fréquence d'échantillonnage doit être 2 fois supérieure à la fréquence du signal pour éviter que les raies créées par l'échantillonnage soient superposées sur les raies du signal).

Dans le cas d'un spectre de largeur infinie (la réalité), il y a donc toujours repliement de spectre. Il est donc nécessaire de filtrer le signal d'origine afin de limiter cet effet de repliement.

Par exemple, dans le cadre de l'audio, on ne va garder que les fréquences que l'oreille est capable d'entendre. Les caractéristiques internes de l'oreille induisent une sensibilité fréquentielle pouvant aller de 20hz à 20khz. C'est pour cette raison que l'on a pris comme fréquence d'échantillonnage  $f_e=44,1$  khz dans le cas du CD.

Ainsi, avant d'échantillonner le signal, on place en amont un filtre qui a pour but d'éliminer toutes les fréquences supérieures à 20khz. C'est un filtre passe bas.



**Figure 10 :** Utilisation du filtre en amont de l'échantillonneur

#### 4.1.4 Modélisation de l'échantillonneur bloqueur

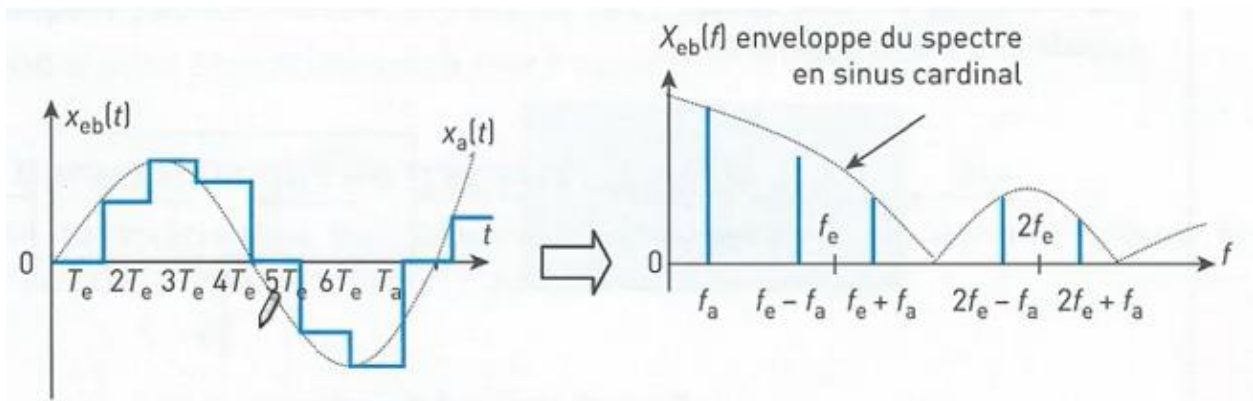
En conclusion, avec un échantillonneur simple (figure 7), le signal d'entrée est prélevé à chaque instant d'échantillonnage donc le codeur nous dispose du signal d'entrée qu'à des instants bien précis où la quantification se devrait faire sur un laps de temps très court et coïncidait exactement aux instants d'échantillonnage. Or, ces deux conditions sont difficiles à réunir.

Par conséquent l'objectif de l'échantillonneur bloqueur est en plus de prélever le signal d'entrée aux instants d'échantillonnage, de maintenir le dernier échantillon à une valeur constante (figure 11). En effet, lors de l'échantillonnage la mesure n'est pas instantanée mais elle dure une certaine période (très courte). Pendant cette période la tension est bloquée et mémorisée à la valeur mesurée. Cela permet à la quantification de disposer de plus de temps pour le codage de l'information.

On suppose le blocage d'une durée  $\theta T_e$  où  $\theta \in ]0 ; 1]$ . L'opération mathématique associée est la convolution du signal échantillonné  $e^*(t)$  avec un rectangle de durée  $T_e$ .

Ce qui alors pour le spectre, revient à le multiplier par un sinus cardinal.

Comme mentionné sur la figure ci-dessous le signal échantillonné sera sur la forme d'un ensemble de points à marche d'escalier, ceci produira une modification sur le spectre où les raies générées n'auront pas la même amplitude que la raie initiale.



**Figure 11 :** Le circuit échantillonneur bloqueur échantillant le signal à des instants bien précis en les bloquant en maintenant pendant la durée ou la période d'échantillonnage : par exemple le 0 est bloqué pendant la durée d'échantillonnage et ainsi de suite concernant le reste des valeurs échantillonnées.

## 4.2 Théorie de la quantification

Le signal échantillonné, bloqué peut à ce stade être converti sous forme binaire (numérique) pour être stocké. Ce codage s'appelle la quantification qui provoque une perte de l'information engendrant une dégradation de la qualité des données (passage de valeurs réels  $x_t$  à des valeurs entières quantifiées et dénombrables).

Le rôle de la quantification est de donner une image binaire d'un signal analogique :

Passage Analogique - Numérique

Signal Continu - Signal discret

Tension - chiffre

### 4.2.1 Principe

A chaque niveau de tension est associé une valeur binaire codée sur  $n$  bits ( $n$  étant la résolution en bit du CAN où plus  $n$  est grand plus le CAN s'approche du signal original) :

$n$  bits vont permettre de distinguer  $2^n$  niveaux de tension répartis de  $-V_m$  à  $+V_m$  (la tension pleine échelle). On a ainsi un pas de quantification qui se calcule par :

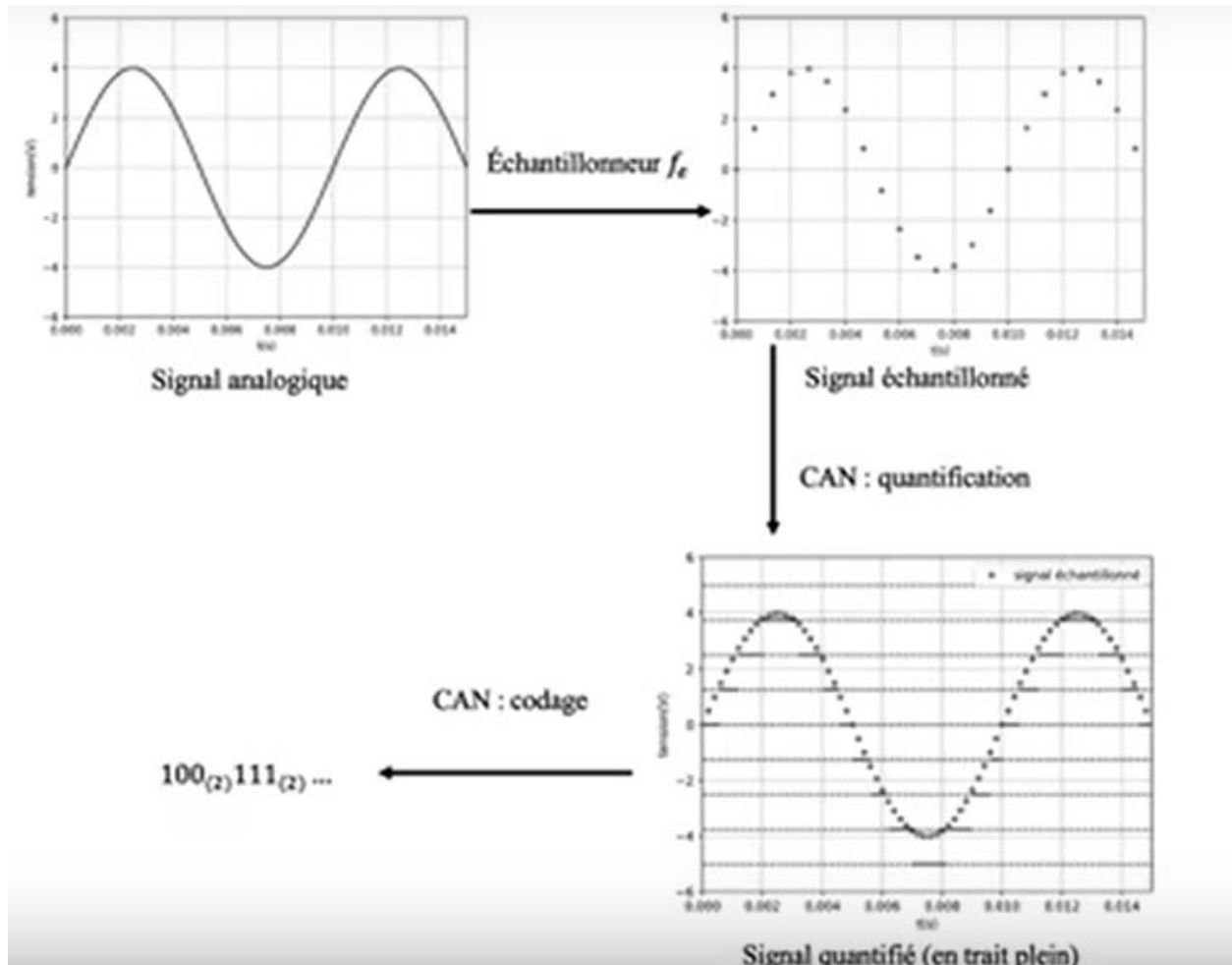
$$q = \frac{2 V_m}{2^n}$$

Ainsi un signal de  $\pm 5V$  codé sur 8 bits donnera un pas de quantification  $q=39mv$ .

La caractéristique d'entrée/sortie d'un CAN est une caractéristique en marche d'escalier.

Chaque palier a une largeur d'un pas de quantification  $q$ .

Le pas de quantification est aussi appelé quantum. Il correspond à la résolution du convertisseur. Le quantum est la plus petite variation de tension que le convertisseur peut coder.



**Figure 12** : nous partons d'un signal analogique continu en temps et en tension ensuite on l'échantillonne en discrétisant le temps selon l'axe des abscisses, le signal échantillonné est bloqué et se verra envoyer dans un CAN pour la quantification du signal en discrétisant les valeurs en tension c'est-à-dire selon les axes des ordonnées. Les valeurs quantifiées seront converties en nombres binaires.

Les étapes de quantification d'un signal :

- 1- Découper l'axe des ordonnées (tension) en parties identiques : nous prenons la tension pleine échelle de 6 à -6 et on la découpe en  $2^n$  valeurs identiques où  $n$  étant la résolution du convertisseur analogique/numérique comme mentionné sur la figure. Nous pouvons aussi distinguer le pas de quantification  $q$  comme mentionné précédemment qui désigne la différence de tension entre deux niveaux de tension quantifiée.

PS : nous utilisons au minimum 8bits lors de la quantification  $\rightarrow 2^8 = 256$  intervalles dans l'échelle de la quantification.

Les ordinateurs des années 1980 avaient un encodage sur 8bits.

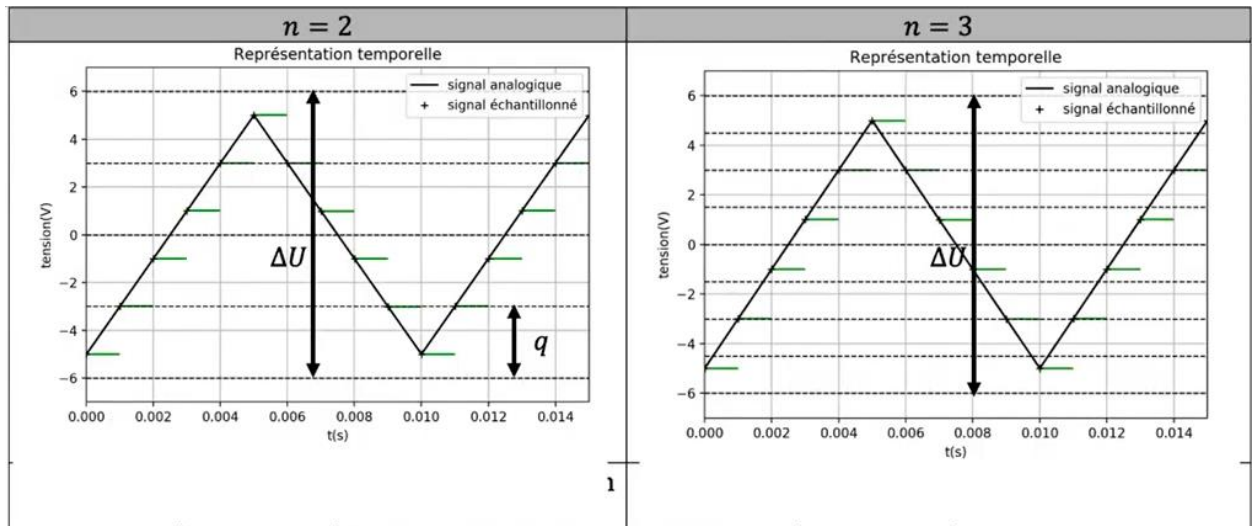
Les CD se comtifiant à 16bits.

Les ordinateurs actuels se comtifiant à 16bits.

Les studios d'enregistrement se comtifiant à 24 ou 32 bits à savoir  $2^{24}$  ou  $2^{32}$  valeurs

➔ une quantification très précise !

- 2- Associer à chaque valeur du signal échantillonné et bloquée (chaque trait vert sur la figure 13) une valeur discrète de tension.



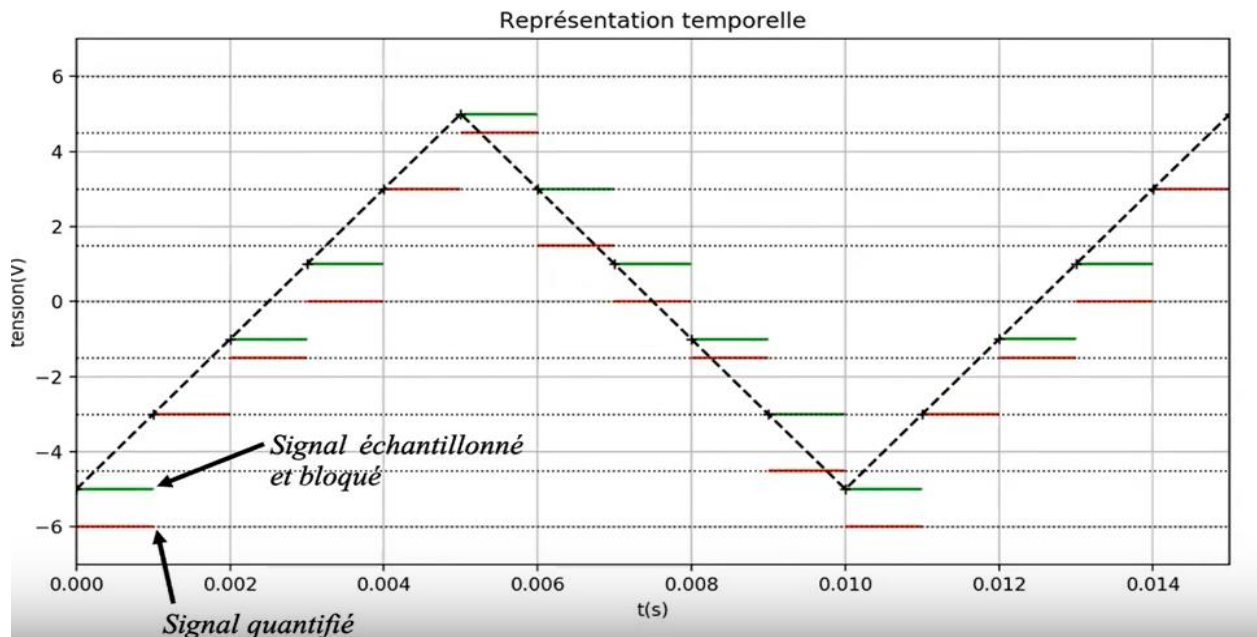
**Figure 13 :** Echantillonnage et quantification d'un signal où le nombre de bits définit le nombre des valeurs quantifiées ( $n=2$  bits donc l'ordonnée sera divisée en 4 segments et si  $n=3$  bits donc l'ordonnée sera divisée en 8 segments)

## 4.2.2 Types de quantifications

### Quantification linéaire

#### 1- Quantification par valeur inférieure

Prenons l'exemple d'un CAN ayant  $n=3$  bits où nous aurons un pas de quantification  $q= 1.5$  V. La quantification par valeur inférieure consiste à attribuer à chaque valeur échantillonnée le niveau de tension quantifié qui est juste en dessous (les paliers rouges sont juste en dessous des paliers verts) comme mentionné sur la figure.

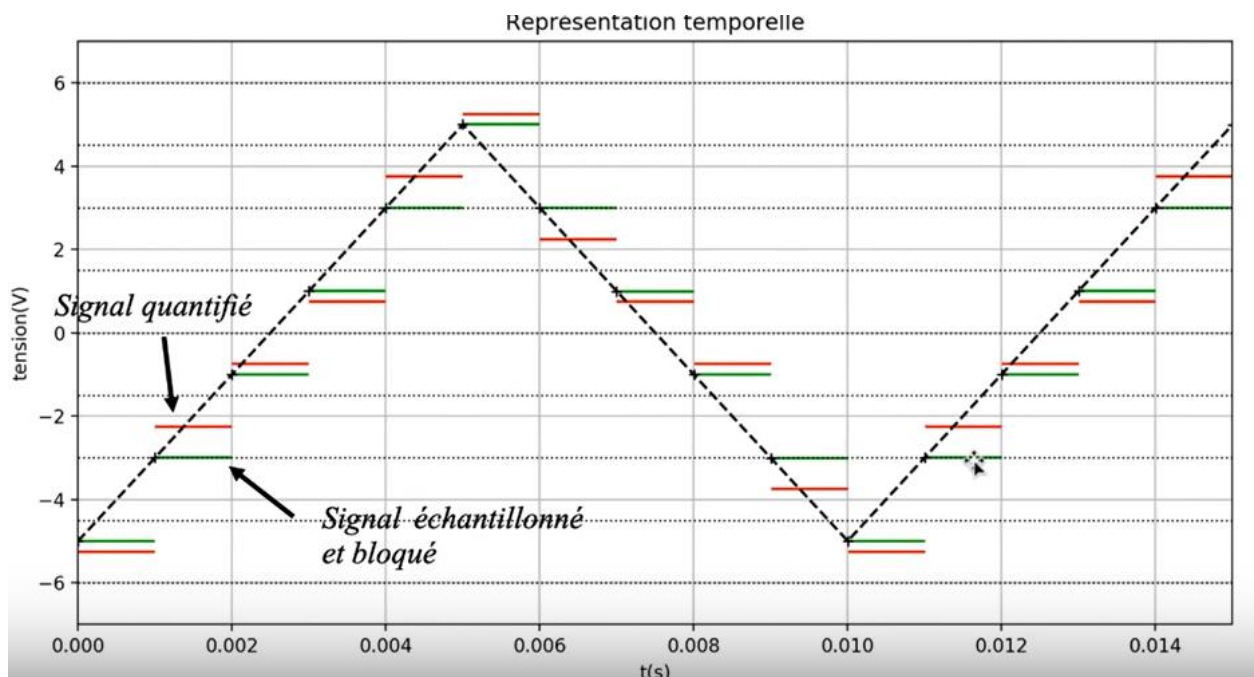


**Figure 14 :** Quantification par valeur inferieur

Il est utile de mentionner que si un signal échantillonné et bloqué se trouve sur un niveau de tension quantifié il reste sur le même niveau comme c'est le cas pour le deuxième et cinquième signal par exemple.

## 2- Quantification par valeur centrale

La quantification par valeur centrale consiste à attribuer à chaque valeur échantillonnée le niveau de tension quantifié moyen comme mentionné sur la figure suivante.



**Figure 15 :** Quantification par valeur centrale

## Rapport signal sur bruit

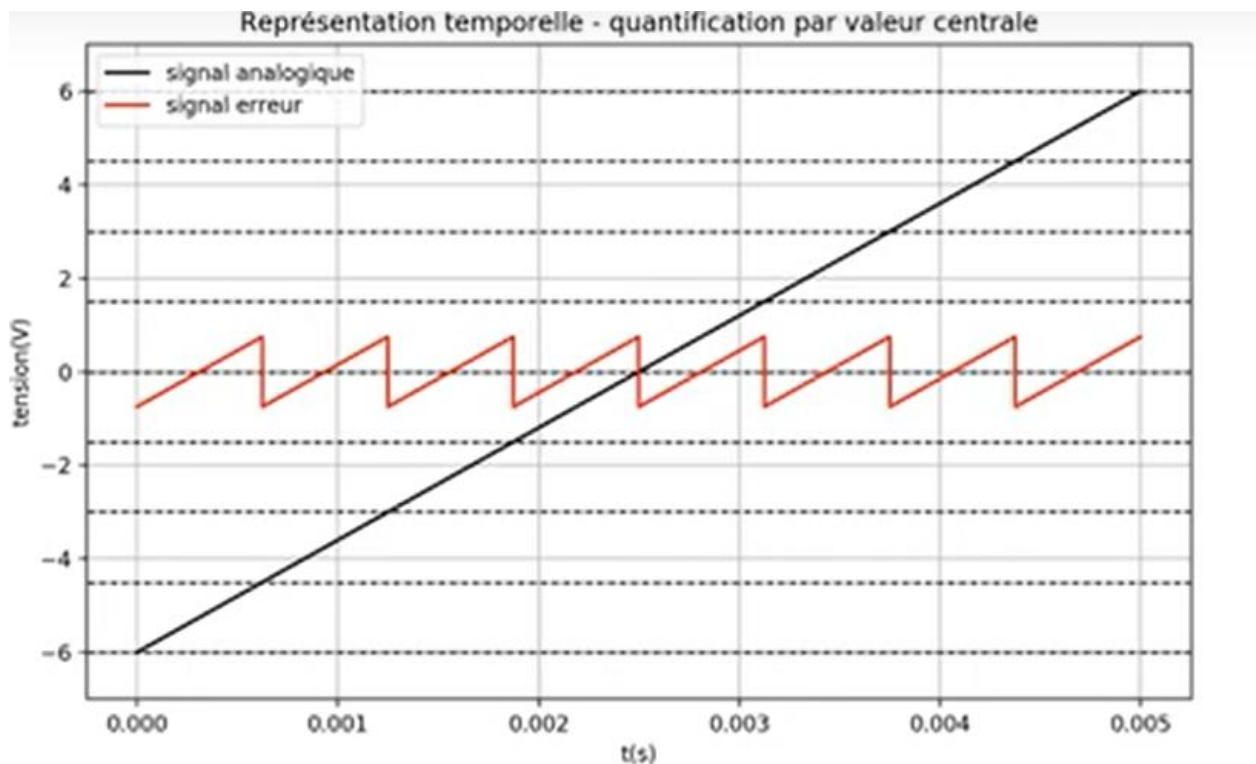
Lorsqu'on réalise la quantification d'un signal un bruit s'ajoutera provenant de l'erreur de quantification du fait que comme indiqué précédemment : pour une valeur bloquée et échantillonnée, nous attribuons une autre valeur (inférieure ou centrale). Nous définissons l'erreur de quantification notée  $b$  comme suit :

$$b = s_e - s_q$$

Où  $s_e$  désigne la valeur échantillonnée et  $s_q$  la valeur quantifiée.

Ceci sera nettement visibles à travers l'apparition d'un bruit de fond par exemple sur deux fichiers sonores où la résolution du CAN est différente (une résolution faible de 8 bits et une autre résolution de 16 bits).

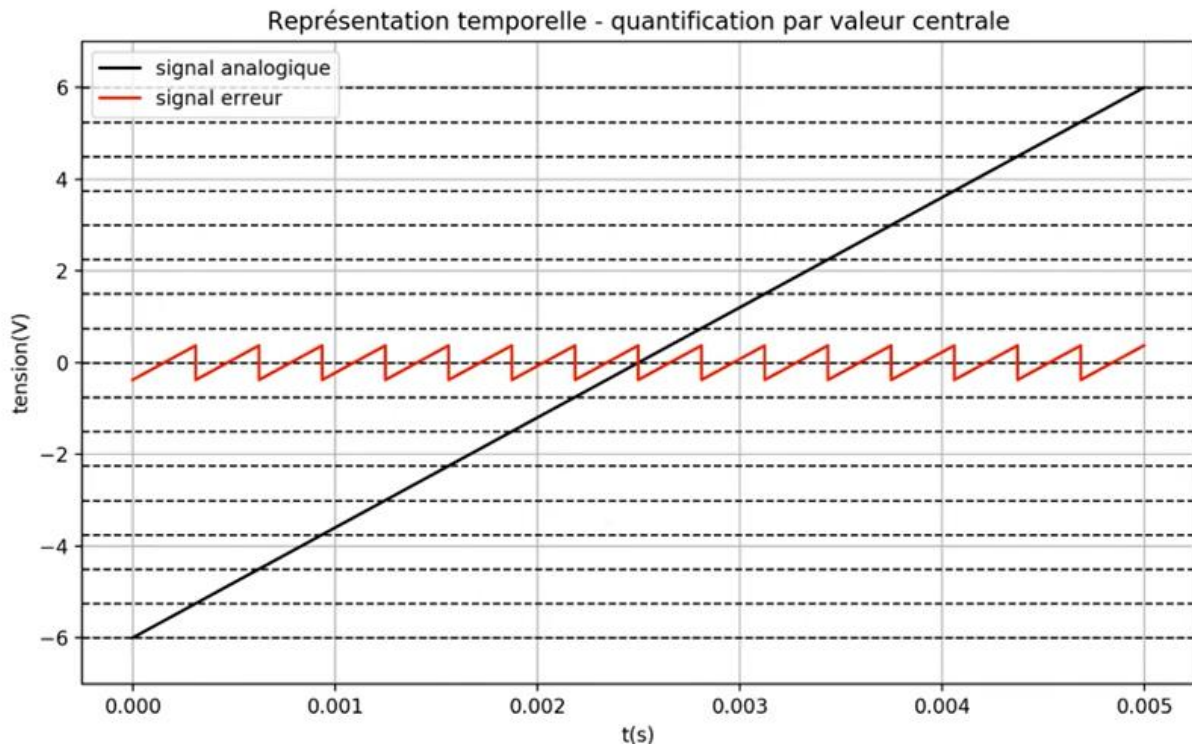
Nous allons présenter sur la figure suivante une demie période du signal triangulaire ( $n=3$  donc une résolution de 8bits). A l'instant  $t=0$  le signal est échantillonné à  $-6V$  avec une erreur qui n'est pas nul vu que la quantification s'est faite par valeur centrale où obligatoirement une erreur apparaisse (changement de la valeur échantillonnée par une valeur centrale quantifiée).



**Figure 16** : Signal erreur d'une quantification par valeur centrale ( $n=3$ )

Nous pouvons affirmer que le signal d'erreur est un signal triangulaire alternatif avec une amplitude égale à la moitié du quantum ( $q/2$ ).

Nous allons présenter sur la figure suivante une demie période du signal triangulaire ( $n=4$  donc une résolution de 16bits).



**Figure 17 :** Signal erreur d'une quantification par valeur centrale ( $n=4$ )

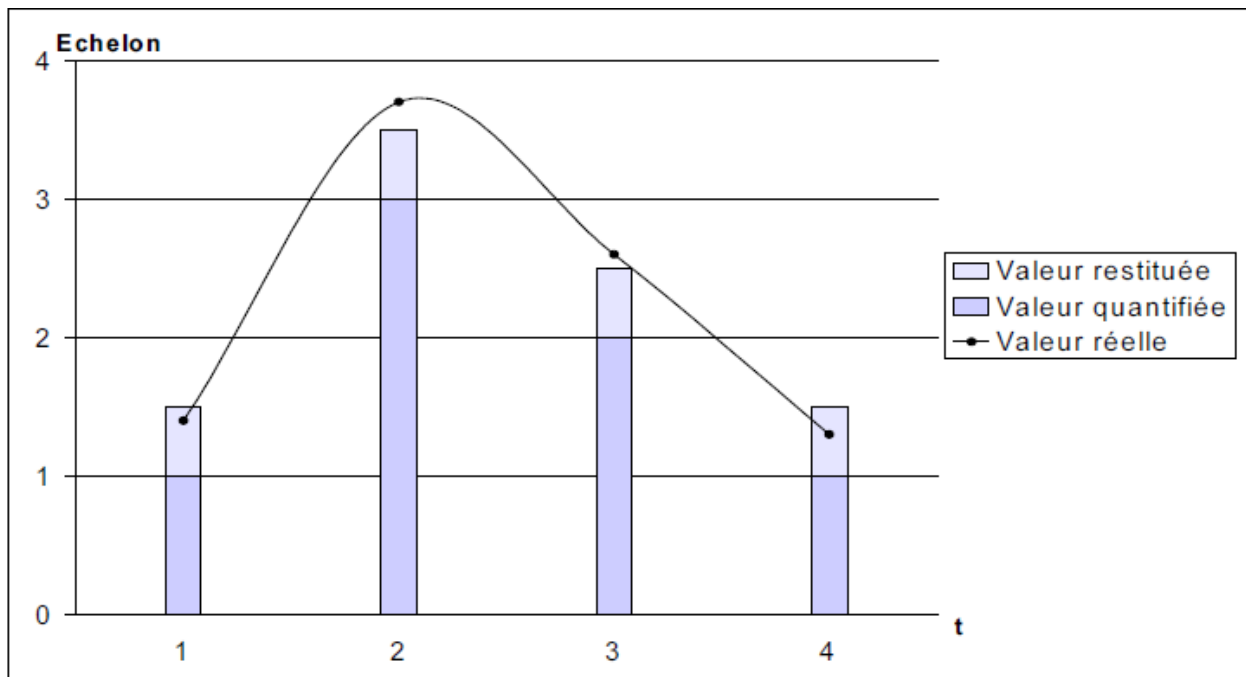
Nous pouvons constater que l'amplitude du signal d'erreur a diminué (toujours de l'ordre de  $q/2$  mais  $q$  a diminué vu l'augmentation de  $n$ ).

### 3- Quantification non linéaire : logarithmique

La bande passante du téléphone est 300 à 3400 Hz, ce qui permet (théorème de Shannon), une fréquence d'échantillonnage  $F_e$  de 8000 Hz, soit une période  $T_e=125\mu s$ .

On suppose au préalable que l'on dispose d'un signal de parole analogique échantillonné à la fréquence de 8000 Hz, c'est-à-dire que l'on a une série de valeurs prises toutes les 125  $\mu s$  sur le signal. Ce sont les points noirs sur la courbe de la figure 18.

Le but de la quantification est de limiter le nombre de valeurs que peut prendre le signal. C'est en effet indispensable, car le nombre de valeurs que peut prendre chaque échantillon doit être fini, puisqu'on utilise un traitement numérique, incompatible avec l'infini. Il faut donc se donner un nombre de bits maximum pour chaque valeur, compatible avec les données à transmettre et le débit voulu. Dans le système MIC (Modulation par impulsions et codage) de représentation numérique d'un signal électrique, le codage se fera sur 8 bits. On va ainsi constituer 256 plages de valeurs ( $2^8$ ), étagées entre  $+V_{sat}$  et  $-V_{sat}$ , auxquels on associera le même code binaire. Par conséquent, tous les échantillons ayant une amplitude comprise dans une même plage seront codés de la même manière. Cette mesure est celle du niveau inférieur de la plage (tension seuil).



**Figure 18 :** Exemple d'une quantification

#### a- Un premier problème

Cependant, on voit rapidement apparaître un premier problème : lors de la restitution du signal, un bruit important est produit. Il est matérialisé par la différence entre la valeur retenue pour le codage et la valeur effectivement échantillonnée. S'il est trop élevé, le signal d'information (la parole) sera inaudible. Pour le réduire, on peut augmenter le nombre des niveaux, mais on se limite en téléphonie à 256 échelons (codage sur 8 bits), soit 128 par polarité, puisque la ligne locale de l'utilisateur ne supporte pas des débits plus élevés que ceux actuellement utilisés en RTC (réseau téléphonique commuté), ce dernier étant imposé par des normes internationales.

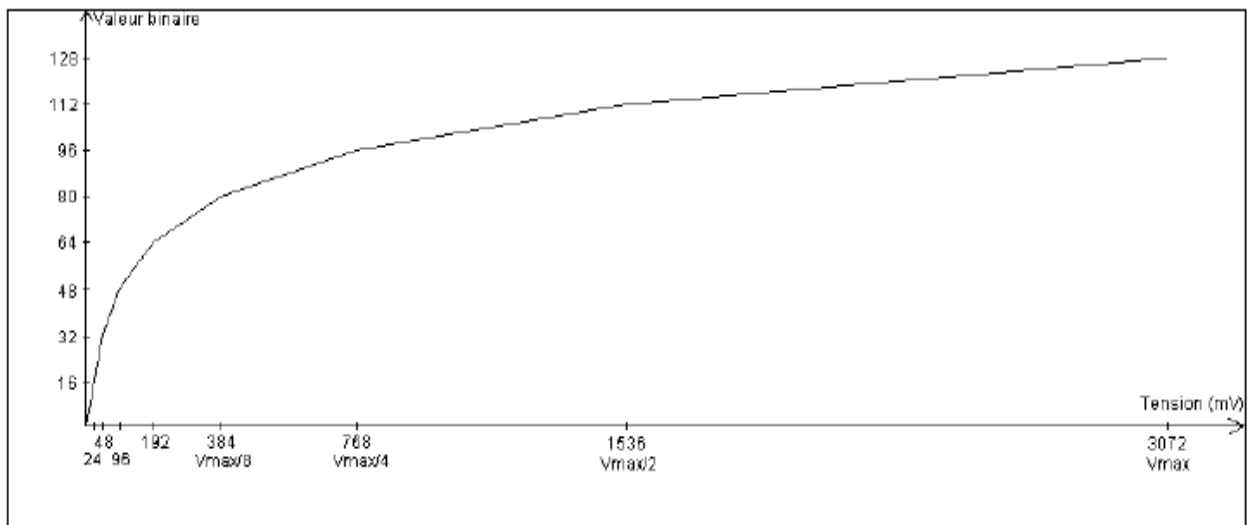
On améliore facilement ce défaut en effectuant ce que l'on appelle une quantification linéaire. Au lieu de restituer la valeur de seuil de la plage, on augmente cette valeur d'un demi échelon (figure 19). Le bruit de quantification est alors limité au plus à  $E/2$ , où  $E$  est l'intervalle de tension entre la valeur seuil et la valeur maximale d'un échelon donné.

Le rapport signal/bruit caractérise le degré d'audibilité de l'information. On rappelle que plus il est élevé, meilleure est la restitution sonore (en ayant une petite valeur pour le bruit le résultat de la division sera grand). Pour référence, le rapport S/B d'un poste radio FM est d'environ 70 dB.

On constate donc ici que S/B n'est pas constant, il augmente avec la tension échantillonnée, et qu'il est trop faible pour les signaux de petite valeur. Ceci ne favorise pas l'intelligibilité de l'information, car il existe un déséquilibre entre les faibles et les forts niveaux, ces derniers ressortant exagérément. De plus la trop grande faiblesse du rapport signal/bruit aux faibles niveaux entraîne une totale intelligibilité de l'information à ces niveaux. Il apparaît par conséquent un deuxième problème à résoudre, sinon la communication sera impraticable.

### b- Un deuxième problème

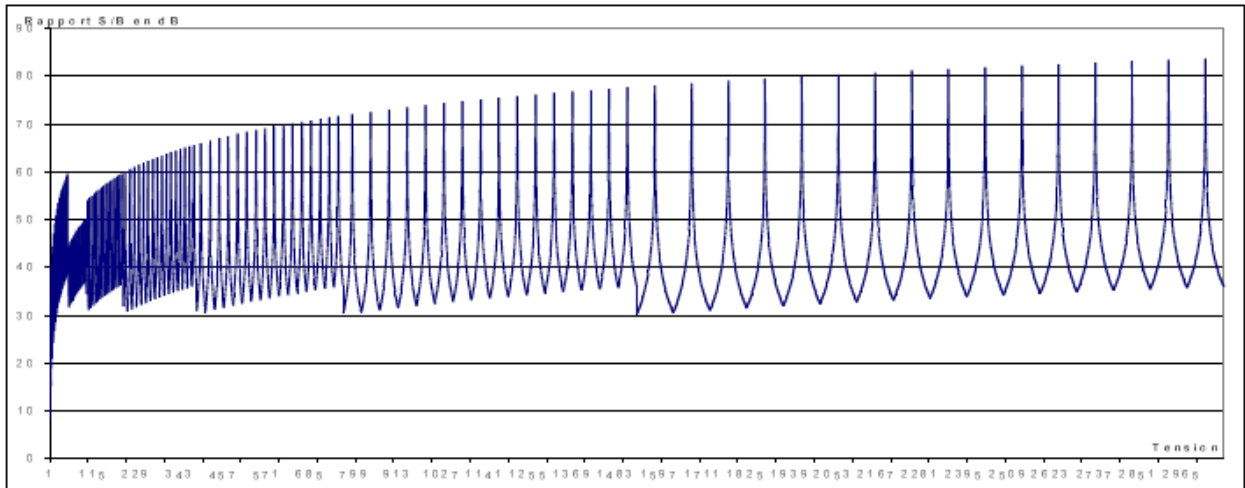
Il est impératif de corriger ce défaut, en trouvant une fonction de quantification qui, toujours étagée sur 256 échelons, va permettre de rendre le rapport signal/bruit constant. Or, augmenter le rapport signal/bruit aux faibles tensions revient à augmenter la précision de la quantification aux faibles niveaux, puisqu'il faut réduire l'erreur. De plus, le rapport signal/bruit étant très important pour les fortes tensions, on peut se permettre de le réduire pour les niveaux hauts. On va donc utiliser une courbe à allure logarithmique, qui offre un pas de quantification réduit aux faibles tensions et plus important aux forts niveaux. Pour cela, on a conçu la loi A. Illustré sur la figure 19 qui est utilisée dans tous les systèmes MIC (le signal est échantillonné puis chaque échantillon est quantifié indépendamment des autres échantillons).



**Figure 19 : La loi A**

La courbe est composée de 8 segments de droite, sur lesquels on pratique une quantification linéaire à 16 plages. Entre 0 et 24mv, un échelon vaut donc 1.5mv ; et entre 1536 et 3072mv, il vaut 96mv.

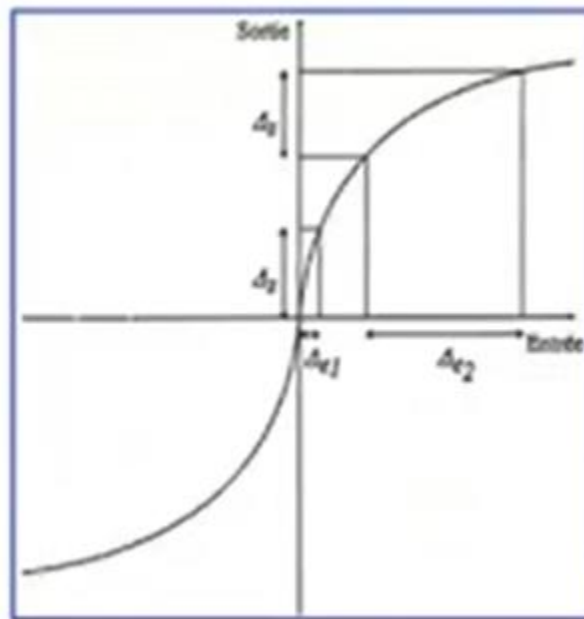
Effectivement, le défaut que possédait la quantification linéaire semble corrigé. Si on mesure le rapport signal/bruit, il est quasiment constant sauf pour les toutes premières plages du premier segment.



**Figure 20 :** Rapport signal/bruit du système MIC

**Le résultat est une meilleure intelligibilité de l'information contenue dans le signal, puisque les bas niveaux ne sont plus masqués par un bruit trop important et le rapport signal/bruit étant à peu près constant, aucun niveau n'est démesurément favorisé par rapport aux autres. Il en résulte une meilleure homogénéité de la parole.**

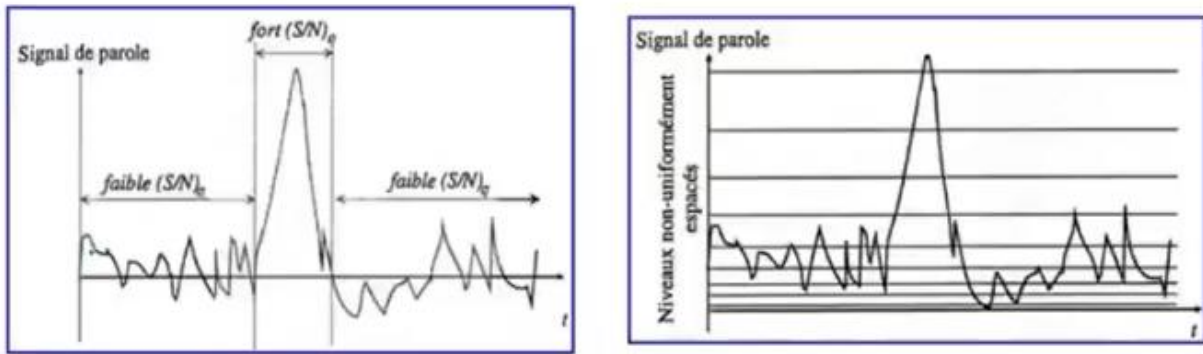
Prenons un autre exemple :



**Figure 21 :** La loi logarithmique

Nous avons un signal en entrée et en sortie, si le signal varie peu sa sortie sera grande et si le signal varie grand sa sortie sera petite. Donc, les faibles amplitudes sont amplifiées  $\Delta_{e1}$  nous génère un  $\Delta_s$  grand et les valeurs assez grandes auront une sortie petite comme le cas de  $\Delta_{e2}$ .

Prenons la figure 22 suivante :



**Figure 22 :**  $q$  dépend de l'amplitude du signal générant une erreur de quantification non constante

Nous pouvons constater que lorsque le niveau du signal est faible, nous devons diminuer le pas de quantification et lorsque le signal est fort, nous l'augmentons. Donc, le pas de quantification dépend de l'amplitude du signal et l'erreur de quantification sera non constante.

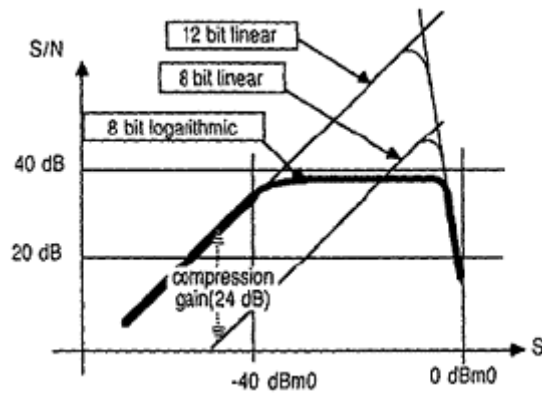
En résumé, la quantification logarithmique permet d'avoir un pas de quantification qui s'adapte en fonction du signal et fut très utilisée en téléphonie mobile selon le principe de compression/expansion pour rendre le signal/bruit constant. Le pas de quantification n'est pas constant et sera différent en fonction du niveau de quantification en choisissant un pas de quantification grand pour les grandes amplitudes du signal et un pas de quantification petit pour les petites amplitudes du signal. L'ultime but est de réduire l'erreur de la quantification vue que le rapport signal sur bruit de quantification dans les quantifications linéaires est plus important que dans la quantification logarithmique.

Les étapes de la quantification logarithmique sont les suivantes :

- 1- Nous prenons un signal d'entrée  $x$  et nous effectuons une compression d'amplitude selon une échelle de quantification logarithmique qui n'est pas linéaire (comme mentionné dans la quantification linéaire) permettant aux grandes amplitudes (proches de l'amplitude maximale) d'avoir un pas de quantification large et celles ayant des valeurs petites d'amplitudes (dues à l'influence du bruit) d'avoir un pas de quantification petit (compression de l'amplitude).
- 2- Nous effectuons ensuite une quantification linéaire comme mentionné précédemment.
- 3- Nous effectuons finalement une expansion d'amplitude (opération inverse).

Dans le cadre de l'audio (cadre de la téléphonie) plutôt que d'utiliser une loi de quantification linéaire, on fait appel à une quantification logarithmique.

L'oreille est un capteur sensible de manière logarithmique au son (une amplitude sonore dix fois plus importante induit une sensation de volume sonore double). On adapte l'amplitude du bruit de quantification à l'amplitude du signal d'entrée **de manière à avoir un rapport signal sur bruit constant (ou presque) et meilleur qu'une quantification linéaire** (courbe ci-dessous).



**Figure 23 :** Evolution du SNR avec l'amplitude du signal

On montre que si la fonction de quantification est une fonction logarithmique du type  $y = 1 + (\log(x)/k)$ , le SNR devient constant et indépendant du niveau du signal d'entrée. Par contre pour les signaux de faible niveau, on ne peut utiliser cette fonction, on utilise une approximation de cette courbe. C'est pour cette raison que l'on trouve deux lois de quantification dans le cas des signaux de la parole :

Loi  $\mu$  : USA/Canada/Japon et philippines

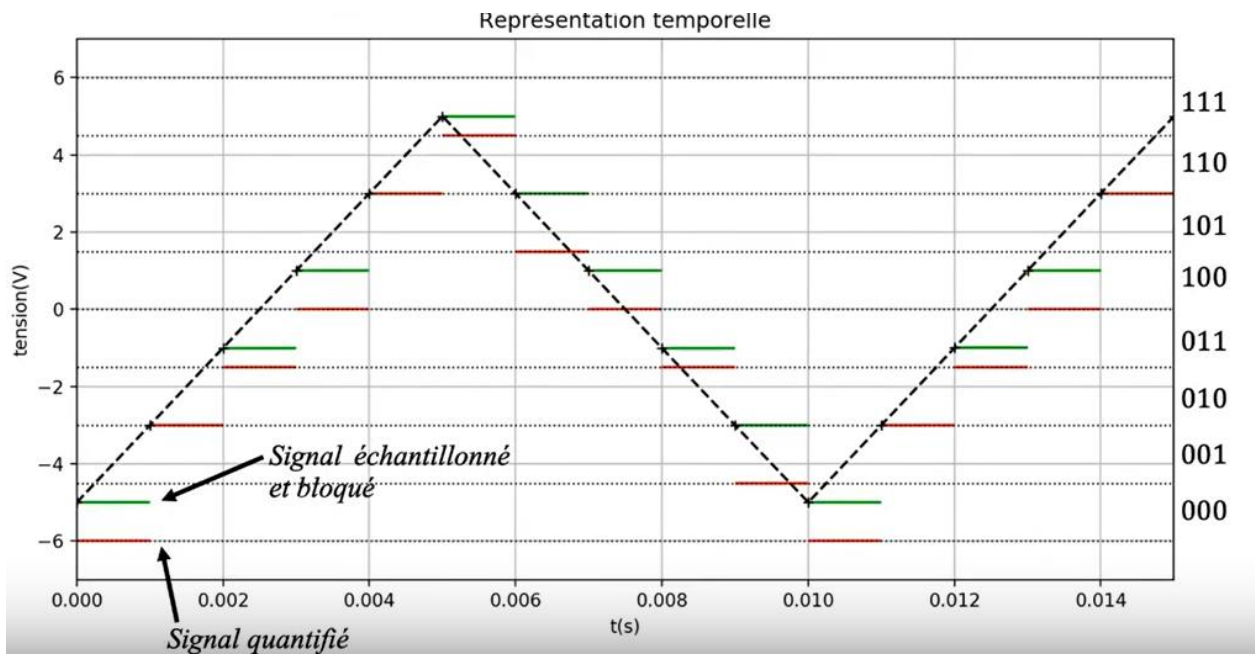
Loi A : Europe et reste du monde

### Choix du nombre de bits de quantification pour le codage

#### Choix classique

Dans le cadre d'une simple acquisition, on peut se contenter de choisir 'n' vis à vis de la résolution souhaitée :

Si nous considérons une règle de quantification par valeur inférieure, donc, par exemple entre le niveau de tension le plus faible -6 et -4.5 nous attribuerons le code 000 comme mentionné sur la figure 24. Dans ce sens, le premier signal sera quantifié et codé comme suit : -5.0V devient -6.0V et le code en sortie sera  $\rightarrow$  000



**Figure 23 :** Exemple d'une quantification par valeur inferieur

## 1. Rappel sur la transformée de Fourier discrète et la transformée en cosinus discrète

La famille des transformées forme le cœur de la compression des données multimédia. En effet, toute transformation cherche à faire ressortir les caractéristiques pertinentes, fondamentales, de l'objet observé. Autrement dit, on cherche à décorréler les données concernant cet objet. S'il s'agit d'un signal audio, les mesures faites à un moment donné dépendent plus ou moins des mesures faites dans le passé. Si c'est un signal vidéo, la dépendance est à la fois spatiale et temporelle. Une information est en relation avec les informations situées au même endroit ou autour de cet endroit qui proviennent d'un passé et d'un futur proches. Cette dépendance est appelée corrélation.

Les transformées utilisées dans le cadre de la compression audio, image et vidéo sont de type fréquentiel. En effet, un signal naturel – c'est-à-dire, mesuré par un capteur – peut être décrit par les fréquences qui le composent. L'analyse est alors une projection du signal sur une famille de fonctions oscillantes de moyenne nulle. Les valeurs de cette projection caractérisent le signal observé. La transformée en cosinus discrète et la décomposition en ondelettes sont les deux outils de la compression [Oppenheim & Schaffer, 2010 ; Gonzalez & Woods, 2018].

Les transformées sont fondamentales. Elles sont l'une des clés d'une compression efficace en termes de débits sur un réseau ou en possibilités de stockage, tout en veillant à la qualité de restitution de l'audio, de l'image ou de la vidéo. Mais, la transformée de Fourier recourant aux nombres complexes n'est pas idéale pour une transformation rapide et efficace d'un signal [Bracewell, 2000 ; Oppenheim & Schaffer, 2010]. Aussi, la transformée en cosinus discrète, qui est sa version à coefficients réels, est primordiale puisqu'elle est utilisée par les standards JPEG et MPEG [Sayood, 2017 ; Watkinson, 2008].

On va scinder l'ensemble des transformées en deux catégories : les transformées spatiales et les transformées fréquentielles. Cependant, toutes ces transformées ont pour but de décrire le signal observé sous un nouveau point de vue afin d'y percevoir plus nettement certaines caractéristiques et propriétés. Elles visent à décorréler le signal. Les mêmes phénomènes apparaissent avec d'autres supports comme, par exemple, l'enregistrement audio d'un discours. Le fait de pouvoir prédire signifie qu'il y a un lien implicatif, une relation temporelle et/ou spatiale, entre les valeurs du signal. On parle alors de signal corrélé. La décorrélation d'un signal a pour but de supprimer ces relations et de mettre en exergue certaines caractéristiques.

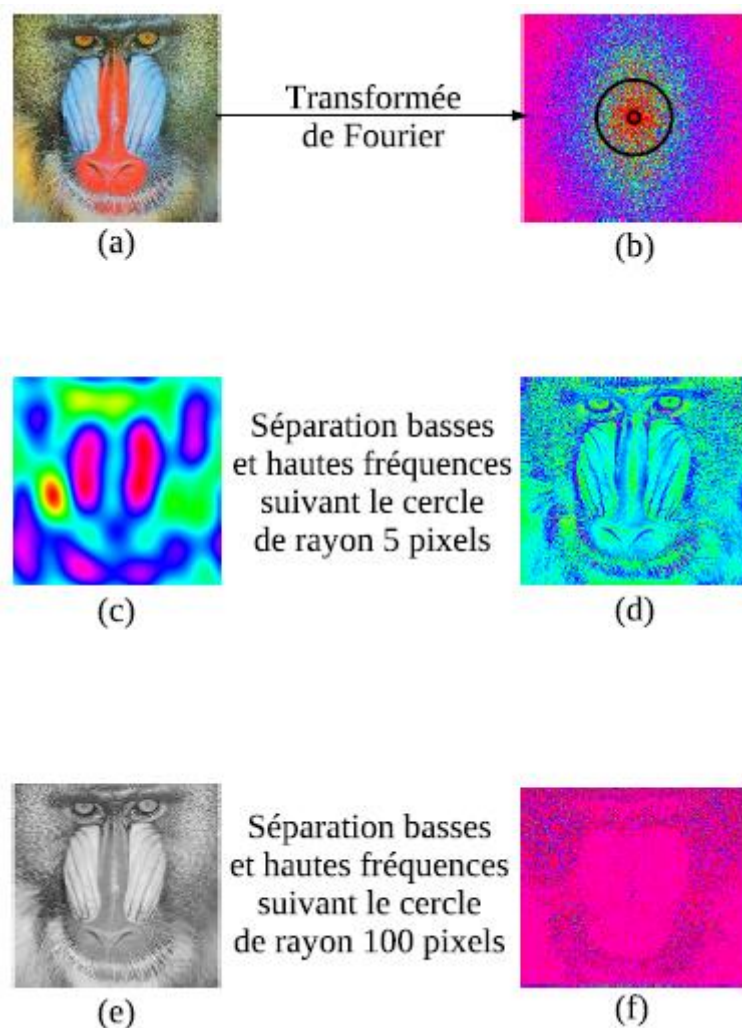
Prenons l'exemple de la figure 1a montrant la photographie (de taille 512x512 pixels) d'un mandrille. La figure 2.b montre le spectre de cette image (l'amplitude de la représentation spectrale). Contrairement, au repère haut gauche habituellement choisi, le repère fréquentiel est situé au centre de l'image ; la fréquence nulle ( $u = v = 0$ ) est au centre de l'image.

On constate aisément que les amplitudes vont en diminuant avec l'augmentation des fréquences. Les fortes amplitudes, indiquées par la couleur blanche sur l'image, sont concentrées autour du centre.

La figure 1c présente les effets de l'annulation des amplitudes pour les fréquences situées au-delà d'un cercle centré à l'origine et de rayon 10. Ce cercle est visualisé sur la figure 1b (le plus petit des deux cercles). La figure 1d présente l'image complémentaire à la figure 1c. C'est-à-dire, l'image reconstruite en annulant les amplitudes des fréquences à l'intérieur du cercle. La sélection des fréquences est sévère. L'image des hautes fréquences est plus fidèle que celle des basses fréquences. L'image originale (figure 1a) est reconstruite à l'identique en additionnant, pixel à pixel, l'image des basses fréquences (figure 1c) et l'image complémentaire (figure 1d).

Le procédé est répété avec un cercle de rayon 100. Maintenant, la quasi-totalité de l'énergie de l'image se trouve dans le cercle (le grand cercle de la figure 1b). Il s'ensuit que la figure 1e est très fidèle à l'image originale. La figure 1f présente l'image complémentaire. Elle est faible en intensité et l'information est très localisée : les hautes fréquences informent sur les détails de l'image observée, alors que les basses fréquences donnent l'allure générale de celle-ci.

A nouveau, l'image originale est reconstruite à l'identique en additionnant l'image de la figure 1e avec l'image de la figure 1f.



**Figure 1.** Illustration par l'exemple de la transformée de Fourier discrète : (a) l'image originale d'un mandrille; (b) son spectre; (c) l'image obtenue par transformation inverse des basses fréquences se

situant dans le petit cercle de l'image (b); (d) l'image complémentaire à l'image (c) obtenue par transformation inverse des hautes fréquences; (e) l'image de la transformée inverse des basses fréquences se situant dans le grand cercle; (f) l'image complémentaire à l'image (e).

La transformée de Fourier présente un inconvénient majeur d'un point de vue purement pratique : elle décrit un signal réel quelconque par un spectre complexe. Ainsi, tout système voulant utiliser ce type de description fréquentielle est amené à manipuler des nombres complexes qui vont augmenter l'espace mémoire occupé ainsi que les erreurs dues aux arrondis de calcul.

Pour ces raisons, la transformée en cosinus discrète (DCT) est préférée à la transformée de Fourier. L'approche est similaire à celle de la DFT mis à part que la phase n'est plus prise en compte. La projection n'est plus une fonction exponentielle complexe, mais, une fonction cosinus qui génère des coefficients réels.

Prenons l'exemple de l'image du mandrille, dont l'analyse fréquentielle – à l'aide d'une transformée de Fourier discrète – a déjà été étudiée.

La figure 2a montre l'image originale. La figure 2b exhibe la représentation spectrale de la transformée en cosinus discrète. Le repère est situé en haut à gauche. Les basses fréquences sont d'amplitudes plus fortes que les amplitudes des hautes fréquences.

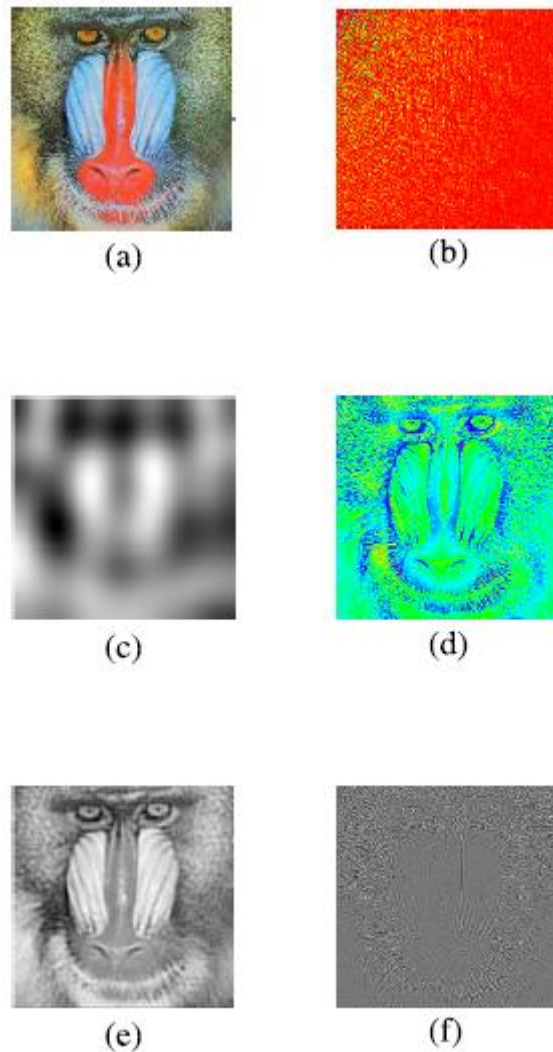
La figure 2c montre l'image du mandrille quand seules sont conservées les dix plus basses fréquences. La figure 2d montre l'image complémentaire obtenue par transformée inverse en annulant les amplitudes des dix plus basses fréquences. D'un point de vue psychovisuel, l'image des hautes fréquences est plus fidèle que l'image des basses fréquences.

L'image originale de la figure 2a est reconstruite à l'identique en additionnant, pixel à pixel, l'image des basses fréquences (figure 2c) avec l'image des hautes fréquences (figure 2d).

Le procédé est répété, mais, en conservant les cent plus basses fréquences.

Les figures 2e et 2f sont les résultats obtenus, respectivement, à partir des basses fréquences et des hautes fréquences. L'approximation, fournie par l'image des basses fréquences, est, maintenant, bien plus fidèle à l'original. Comme avec la transformée de Fourier, l'image des hautes fréquences montre les détails supprimés de l'image d'approximation. Ces détails sont très localisés. En ajoutant l'image de la figure 2f à l'image d'approximation, l'image originale est reconstruite à l'identique.

En conclusion, la DCT se comporte de manière analogue à la transformée de Fourier. Son avantage réside dans l'utilisation de coefficients réels et non complexes.



**Figure 2.** Illustration par l'exemple de la transformée en cosinus discrète : (a) l'image originale d'un mandrille; (b) son spectre; (c) l'image obtenue par transformation inverse en ne conservant que les dix plus basses fréquences; (d) l'image complémentaire à l'image (c) obtenue par transformation inverse des hautes fréquences; (e) l'image de la transformée inverse des cent plus basses fréquences; (f) l'image complémentaire à l'image (e).

## 2. Définition d'une vidéo

Le mot vidéo vient du latin vidéo qui signifie « je vois ». C'est un terme qui regroupe l'ensemble des techniques permettant la visualisation ou l'enregistrement d'images animées accompagnées de son, sur un support électronique. Une vidéo est une succession d'images à une certaine cadence. L'œil humain est capable de distinguer environ 20 images par seconde. On caractérise la fluidité (vitesse) d'une vidéo par le nombre d'images par secondes (en anglais frame rate), exprimé en FPS (Frames per second).

### 2.1 L'acquisition des images par une caméra

### 2.1.1 Principe de fonctionnement d'une caméra vidéo

#### a) Principes de base.

Une caméra vidéo est constituée d'un objectif permettant de focaliser les images de la scène à enregistrer, ainsi que d'une chambre noire dont le fond est occupé par une surface photosensible comme mentionné sur la figure 3 [Szeliski, 2022].

Selon les matériels utilisés, la surface photosensible peut être un film argentique ou un capteur photosensible électronique. Dans le second cas, le capteur est constitué d'une matrice de pixels qui constituent la cible où se forme l'image.

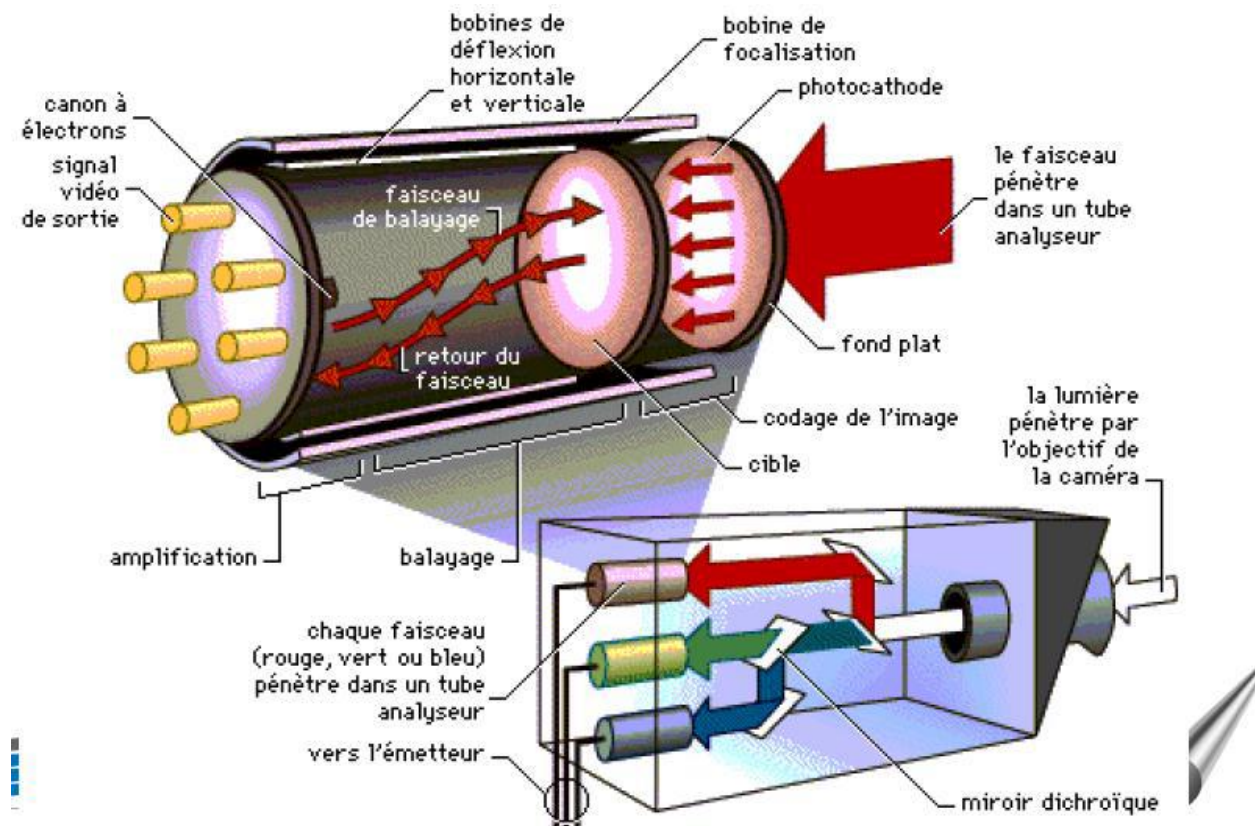
Lorsque l'une des cellules de la matrice est éclairée, elle crée alors un courant électrique proportionnel à l'éclairement qu'elle reçoit. Tous les courants émis par les cellules sont ensuite traités par des circuits électroniques de la caméra afin de reconstituer l'image sous la forme d'un signal vidéo, puis de l'enregistrer sur la bande.

#### b) Fonctionnement d'un capteur CCD.

Dans le but de produire des images compatibles avec les téléviseurs, celles-ci sont généralement constituées de demi-images entrelacées. A raison d'une cadence de 25 images par seconde, une demi-image est ainsi focalisée sur le capteur CCD tous les 1/50<sup>e</sup> de seconde.

La surface du capteur est constituée de cellules photosensibles recouvertes chacune d'un filtre coloré, afin de mesurer de façon précise l'intensité lumineuse d'une couleur primaire et ainsi permettre la reconstitution d'images couleur.

Chaque impulsion produite par une cellule excitée est généralement codée sur un octet, ce qui permet de reproduire près de 16,7 millions de couleurs différentes après combinaison des trois couleurs primaires. Bien qu'élevé, ce nombre peut toutefois s'avérer limité dans les situations de sous-exposition et de surexposition, ce qui explique que certains appareils gèrent des couleurs d'une profondeur plus forte (codage sur 30, 36 voire 48 bits).



**Figure3.** Fonctionnement d'une caméra vidéo

### 2.1.2 Les formats de transmission et d'enregistrement

Les caméras utilisent des capteurs électroniques permettent la transduction de la lumière en courant électrique (conversion d'une information optique en une information électrique). A la sortie du capteur couleur, les trois signaux fondamentaux que sont le rouge, le vert et le bleu sont séparés. Les différents formats de transmission et d'enregistrement diffèrent par le mode de combinaison de ces trois signaux [Bovik, 2005].

- Format RGB:

Les trois informations de base sont véhiculées par des câbles séparés (qualité optimale mais appareils coûteux et poids élevé).

- Format composante:

Les informations de luminance (Y) et les informations de chrominance (R-Y et B-Y) sont transmises et enregistrées sur des pistes différentes. Lors de l'enregistrement, on diminue la quantité d'information concernant la couleur (R-Y et B-Y), mais aucune perte de qualité ne sera visible puisque la compression portera principalement sur les informations de chrominance inutilisées par l'œil humain. Ce format garantit ainsi une qualité irréprochable.

- Format S-vidéo:

Le format S-vidéo (Sony-vidéo) transmet les informations de luminance et de chrominance via deux câbles différents. Les signaux de luminance et de chrominance sont ensuite enregistrés sur une même piste magnétique.

·Format Composite:

Les informations de luminance et de chrominance sont combinées en un seul signal (VHS, 8 mm, U-matic...), ce qui peut entraîner des effets de moirage. Comme pour le format précédent, les signaux de luminance et de chrominance sont enregistrés sur une même piste magnétique.

### **3. La numérisation d'un signal vidéo analogique**

#### **3.1 Généralités**

Afin de numériser une vidéo analogique, il faut diviser chaque image vidéo selon une résolution donnée (généralement 720 x 486) puis associer une valeur numérique à chacun des éléments qui forment la couleur de ce pixel (RGB, YUV) en utilisant une table de conversion des couleurs (en général 24 bits par pixel pour 16,7 millions de couleurs possibles en chaque point).

Etant donné qu'une image vidéo contient plusieurs milliers de pixels et que la vidéo analogique NTSC défile à plus de 30 images par seconde (25 images de 720 × 576 pixels par seconde en PAL), il est nécessaire d'utiliser des puces performantes permettant une conversion en temps réel.

#### **3.2 La norme 4 :2 :2**

Pour convertir un signal analogique en signal numérique, il faut procéder à une étape d'échantillonnage qui détermine des tranches temporelles puis effectue une quantification qui consiste à prélever ponctuellement la valeur du signal à des instants réguliers (période de découpage).

##### **3.2.1 L'échantillonnage**

Afin d'échantillonner un signal en préservant son information, il est indispensable que la fréquence d'échantillonnage soit  $> 2 F_{max}$  (loi mathématique de Shannon et Nyquist). Dans le cas contraire, les composantes spectrales répétitives du signal échantillonné peuvent se chevaucher et donner naissance à une fréquence parasite produisant un effet de moiré à l'image. Un filtre passe-bas disposé en amont de la conversion permet d'éviter cette situation.

Le signal vidéo de luminance possède une bande passante d'environ 6 MHz, ce qui implique que pour être justement numérisée,  $F_e(Y) > 2 \times 6 = 12$  MHz (la fréquence d'échantillonnage doit être supérieure à 2 fois la fréquence maximale). Ce qui explique le choix de 13.5 MHz.

L'œil humain étant moins sensible aux couleurs qu'à la luminosité, la fréquence d'échantillonnage de la chrominance a été établie à 6.75 MHz.

Pour un groupe de 8 pixels (2 lignes de 4 pixels chacune), le chiffre 4 de la norme 4 :2 :2 représente le nombre de valeurs de luminance par ligne (13,5 MHz). Les deux valeurs égales à 2 désignent quant à elles respectivement le nombre de pixels des lignes paires puis des lignes impaires qui possèdent une valeur propre de chrominance ( $6,75 \text{ MHz} = 13,5/2$ ).

4 :2 :2 → le niveau de sub-sampling ou de sous-échantillonnage

Les principaux CODEC actuels compressent la chrominance car l'œil humain est sensible à la luminance (durant la nuit il nous suffit un peu de lumière pour apercevoir tout) et moins à la chrominance (durant la nuit on ne distingue pas parfaitement les couleurs). En effet, la compression est qualitative et n'est pas perceptible car elle est adaptée à notre physiologie pour ne pas devenir perceptible.

### 3.2.2 La quantification

Un quantificateur permet de convertir les tensions en valeurs numériques mais il provoque une erreur d'arrondi car alors qu'un signal analogique peut prendre une infinité de valeurs, le signal numérique est codé sur un nombre  $n$  de bits. La précision du signal converti est donc liée au nombre de valeurs disponibles pour traduire chaque échantillon. (L'intervalle situé entre deux valeurs se nomme le pas de quantification).

La quantification d'un signal vidéo est uniforme, linéaire. Initialement fixée sur 8 bits, la quantification de la norme 4 :2 :2 a été passée à 10 bits pour accroître le nombre de niveaux numériques disponibles et éviter les imperfections dans le rendu des dégradés.

### 3.2.3 Le codage

Une fois échantillonné et quantifié, le signal vidéo doit être codé afin d'optimiser son stockage ou sa transmission [Sayood, 2017].

- Le code NRZ (Non Retour à Zéro) : une donnée binaire "1" engendre un niveau haut de signal et une donnée "0" un niveau bas.
- Le code NRZI (Non Retour à Zéro Inversé) : une donnée binaire unitaire engendre une transition au milieu de la demi-période d'horloge alors qu'une donnée nulle n'a aucun effet.

Ce type de codage est utilisé en vidéo dans les liaisons série 4 :2 :2 car il permet de transmettre le signal vidéo avec son signal d'horloge.

- Le code Biphase Mark : une donnée nulle provoque une transition et un maintien du niveau pendant toute la période d'horloge, alors qu'une donnée unitaire entraîne une transition et un changement de niveau à la moitié de la demi-période d'horloge.

### **3.2.4 Détection et correction des erreurs**

Le support d'enregistrement ou le canal de transmission peuvent engendrer des erreurs dans le flux de données numériques, ce qui a conduit à développer des systèmes de correction. Afin que ces systèmes puissent détecter les erreurs, des données redondantes sont ajoutées lors du codage aux données utiles, selon une loi définie et connue du codeur et du décodeur.

Chaque fois que cette loi n'est pas vérifiée au décodage, un processus de correction est automatiquement déclenché. Dans le cas où le manque d'information est tel que même les données redondantes ne suffisent pas pour retrouver le signal d'origine, des processus de compensation qui calculent la valeur moyenne entre des échantillons proches sont alors exécutés.

## **4. Les images numériques de la vidéo**

### **4.1 Les images vidéo non compressées**

Si un signal vidéo de 720 x 486 pixels de résolution est numérisé en utilisant la norme YUV 4 :2 :2, le fichier résultant sera de 683.44 Ko par image ou 20.02 Mo/sec. C'est ce qu'on appelle le format non-compressé.

Avec un tel débit, il faudrait 1.2 Go d'espace sur une mémoire de masse pour stocker une seule minute de vidéo, ce qui amène à s'interroger sur les possibilités de réduire la taille des fichiers. De plus, il faut également faire face au problème du transfert des données digitales.

### **4.2 La réduction du débit**

De manière à réduire au maximum le nombre d'octets utilisés pour représenter une image et ainsi réduire le débit binaire nécessaire pour la transmettre, plusieurs solutions ont été développées.

- Diminution du nombre d'images par seconde (en dessous de 15 à 18 images/s, l'œil perçoit des saccades)
- Réduction du nombre de points de l'image.

Réduction du nombre d'informations de couleurs codées (2 octets par pixel soit 64000 couleurs voire 1 octet par pixel pour 256 couleurs) : solution acceptable pour le multimédia mais à proscrire pour le montage vidéo.

Bien que toutes ces méthodes permettent de réduire le débit des données, elles provoquent une très forte dégradation des images, ce qui a conduit à rechercher un autre moyen pour diminuer la taille des fichiers. La compression (réduction du débit binaire) des fichiers a alors été développée pour atténuer la perte de qualité.

## **5. La compression de la vidéo**

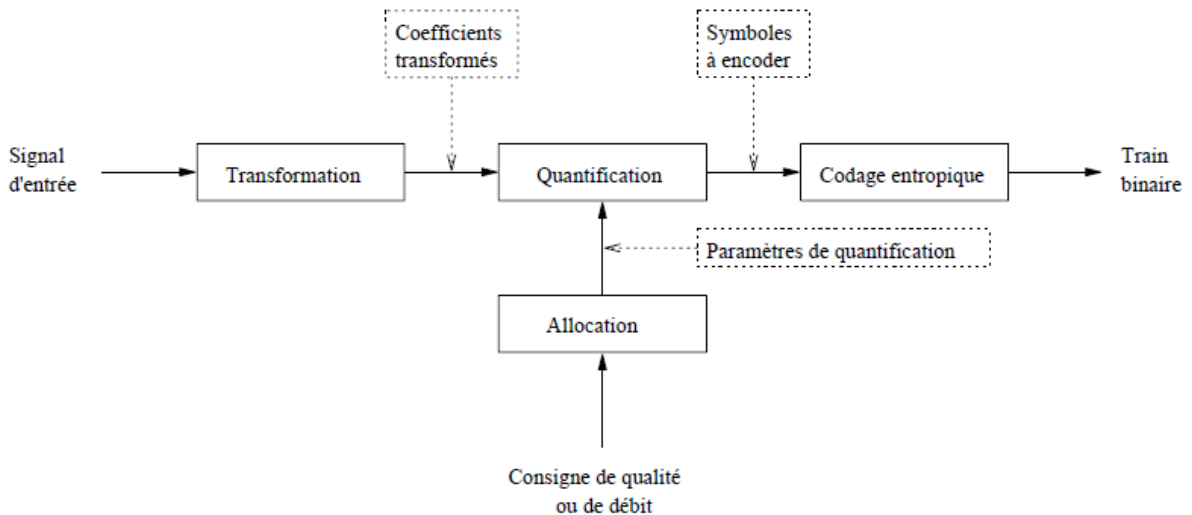
Cette compression est basée sur l'observation selon laquelle une image contient de nombreuses informations redondantes [Watkinson, 2008 ; Sayood, 2017].

- Redondance spatiale (intra-image) : dans une image, deux points voisins sont souvent similaires → les approches utilisées pour la compression intra-image sont : transformation des couleurs (ex : de l'RVB à l'YUV), JPEG utilisée par MPEG.
- Redondance temporelle (inter-image) : deux images successives présentent souvent de nombreuses similarités.

Les approches utilisées pour la compression inter-image sont : le codage différentiel où une image est comparée à une image de référence et seules les pixels qui ont changé par rapport à cette image de référence seront codés, la compensation de mouvement [Sayood, 2017] tenant compte du fait qu'une bonne partie qui compose une nouvelle image dans une séquence vidéo se retrouve dans l'image précédente mais pas forcément au même endroit donc il est possible de prédire une nouvelle image en cherchant un bloc identique bloc par bloc sur une image de référence et si un bloc référent est trouvé l'encodeur code l'emplacement de l'image référence où se situe le bloc identique ainsi que le vecteur de mouvement qui utilise moins d'espace que le contenu réel d'un bloc.

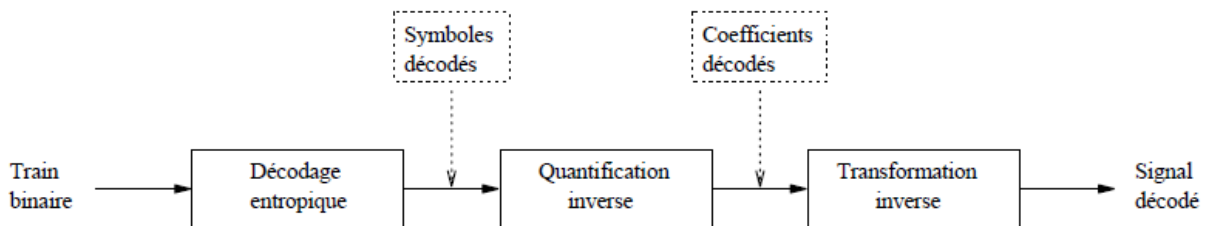
La compression détecte les redondances et les élimine. Toutefois, il est indispensable que ce processus n'empêche pas de reproduire l'image originale de façon intacte ou tout du moins très proche.

Nous présentons sur la figure suivante le schéma général de compression des images et des vidéos.



**Figure 4.** Schéma général de compression des images et des vidéos.

Quant à la figure suivante elle présente le schéma général de la décompression des images et des vidéos.



**Figure5.** Schéma général de décompression des images et des vidéos.

Si l'on choisit correctement le type d'informations qui vont être dégradées, il est possible de reconstruire des images que l'œil humain ne pourra pas différencier des originales. En effet, celui-ci ne perçoit que peu une dégradation des couleurs dans une scène remplie d'objets en mouvement rapide tandis qu'une faible perte de qualité dans une image fixe comportant un dégradé de couleurs sera immédiatement perçue.

La vidéo numérique utilise donc les particularités de l'œil humain afin de proposer des résultats visuellement sans perte. Diverses compressions ont été mises au point :

- Variable length coding : il est basé sur le fait que certaines combinaisons de pixels sont plus fréquentes que d'autres.

Il recense toutes les combinaisons possibles d'un nombre donné de pixels puis détermine leur fréquence d'apparition dans une image. Il attribue ensuite un code à chaque combinaison dont le nombre de bit est d'autant plus réduit que la combinaison est fréquente.

- Differential pulse code modulation (DPCM) : dans la plupart des images, la différence entre deux pixels adjacents est souvent faible (les transitions franches sont rares), ce qui permet de prédire la valeur d'un pixel en connaissant l'un de ses voisins.

- **Discrete cosine transform (DCT)** : cette méthode se base sur la transformation d'une représentation spatiale d'un bloc de pixels (position horizontale, verticale et amplitude), en une représentation sous forme mathématique différente qui requiert de traiter moins d'informations. En effet, cette nouvelle représentation ne se base plus sur une analyse spatiale mais sur une analyse fréquentielle soigneusement calculée.

## 5.1 Présentation de quelques standards de compression

- JPEG (joint photographic experts group):

Standard international pour la compression d'images couleurs fixes stockées sous forme digitale. Il accepte n'importe quelle définition et autorise un nombre de bits par pixel compris entre 1 et 32.

La technique de compression utilisée est basée sur la Discrete Cosine Transform (DCT).

Cependant, conçu pour les images fixes, il ne convient pas pour la compression de séquences vidéo.

- Motion JPEG:

Extension du JPEG qui permet de traiter des séquences d'images. Il considère une vidéo comme une succession d'images fixes qu'il compresses en utilisant le standard JPEG.

Cette technique permet un montage à l'image près. De plus, si l'on se contente de facteurs de compression compris entre 2 :1 et 4 :1, la dégradation est négligeable.

Toutefois, lorsque la compression est trop forte, les images deviennent visuellement inacceptables.

- MPEG (moving pictures experts group):

Si l'on souhaite conserver un niveau de qualité compatible avec une utilisation professionnelle, la compression par images séparées ne peut pas dépasser un facteur de 4 :1.

Aussi, afin de permettre des taux de compression supérieurs sans pour autant obtenir des images de piètre qualité, le MPEG qui traite les similitudes qui existent entre plusieurs images successives a été mis au point.

A l'origine, ce standard prévoyait 4 niveaux [Watkinson, 2008 ; Richardson, 2010]:

- MPEG-1 : destiné aux applications multimédia (produire des images de qualité équivalente au VHS en descendant à un débit binaire de l'ordre de 1.2 Mbits/seconde (1.5 Mbits/seconde en incluant le son).
- MPEG-2 : extension de MPEG-1 permettant d'obtenir une qualité d'image supérieure par traitement de séquences d'images entrelacées. (Le but était de produire des images de la qualité d'un système vidéo composite avec un débit binaire de l'ordre de 4 à 8 Mbits/seconde ou des images de haute qualité avec un débit de 10 à 15 Mbits/seconde.)
- MPEG-3 : destiné à la télévision haute définition. Cependant, en raison de la performance de MPEG-2, le MPEG-3 n'a pas été développé.
- MPEG-4 : destiné aux communications mobiles, il n'a rien à voir avec le monde de la vidéo broadcast.

## 5.2 MPEG1

Le MPEG-1 (développé en 1993) permet d'obtenir des débits de l'ordre de 1,2 Mbit/s, ce qui le rend exploitable par un lecteur de CD-ROM. En effet, les vidéos à cette époque étaient de l'ordre de 352 x 240 ou 352 x 288 selon 25 ou 30 images / seconde avec un audio track de 1.4 megabits /seconde.

Afin de permettre un accès aléatoire, la recherche en vitesses avant (FF = fast forward) et arrière (FR = fast reverse), cet algorithme introduit le concept d'images prédictees et interpolées bidirectionnellement.

Trois types de trames sont considérés :

- Trames I (intra coded frames = codage interne) : elles sont codées sans aucune référence à une autre image de la séquence vidéo. Elles permettent l'accès aléatoire ainsi que les fonctionnalités FF et FR mais le taux de compression est très bas
- Trames P (predictive coded frames) : elles sont codées avec une référence à l'image précédente et ne permettent pas l'accès aléatoire et les fonctionnalités FF/FR. (Elles sont utilisées pour la prédiction de trames futures ou passées)
- Trames B (bidirectionally predictive coded frames) : elles ont besoin des trames futures et passées comme référence pour être codées. Elles sont utilisées pour obtenir un fort taux de compression mais ne peuvent pas être utilisées comme référence.

Une combinaison de ces trois types de trames permet un taux de compression élevé tout en autorisant un degré d'accès aléatoire et de FF/FR raisonnable. Néanmoins, elle nécessite un temps de codage élevé, ce qui peut être un handicap dans le cadre de la visiophonie et de la vidéoconférence.

Les images sont décodées en faisant des moyennes par blocs (DC coded frames).

Le MPEG1 (comme les MPEG2 et 4), comporte plusieurs parties dont la partie vidéo (part2) et la partie audio (part3). La couche audio se divise également en trois couches de complexité et d'efficacité de compression différente. La couche MPEG1 audio layer 3 a donné naissance au format audio MP3.

Actuellement, ce standard est progressivement abandonné au profit du MPEG2.

### **5.3 MPEG2**

Cet algorithme est, au départ, identique à celui de MPEG1. Toutefois, il a été complété par d'autres algorithmes afin de permettre un traitement des images entrelacées. Le MPEG2 visait des usages liés uniquement à la télévision numérique avec une résolution de 720 x480 ainsi qu'aux DVDs.

Notons qu'également le MPEG2 inclut le format audio MP3, cependant, les DVDs et les télévisions numériques n'utilisent pas le MP3 comme un codec audio.

### **5.4 MPEG4 : soutenu par l'ISO**

Contrairement au MPEG2 qui est limité à la télévision numérique, le MPEG4 englobe de nombreuses applications multimédia (téléchargement et streaming sur internet, multimédia sur mobile, radio numérique, jeux vidéo, la télévision, supports haute définition...)

Il se décompose en plusieurs parties elles-mêmes constituées de plusieurs profils (collection d'algorithmes) et de niveaux (contraintes quantitatives).

MPEG4 ASP → implémenté dans DivX et XViD.

MPEG4 AVC ou H.264 reconnu comme le standard actuel pour des vidéo en haute définition.

Notons que la majorité des approches de compression de la vidéo apparue jusqu'à 2020 utilisent la DCT ou la transformée discrète en sinus (DST) comme une partie du processus de compression.

## **6 Détails du codage MPEG**

### **6.1 Le codage spatial**

Dans un premier temps, il est nécessaire d'effectuer une analyse de fréquence spatiale à l'aide d'une transformée (outil mathématique qui permet de traduire une forme d'onde en différents domaines). Le résultat d'une transformée est une suite de coefficients décrivant l'amplitude de chaque composante fréquentielle présente dans le signal. Si les coefficients sont gérés avec

une précision suffisante, la sortie de la transformée inverse doit être identique à la forme d'onde originale.

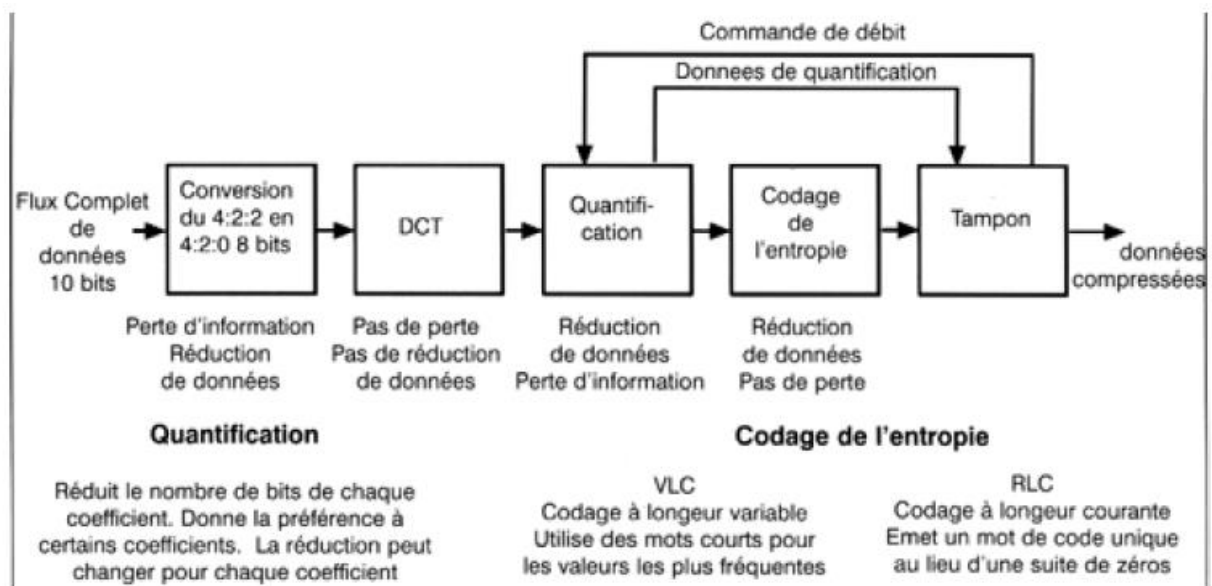
La transformée cosinus discrète (DCT) est une version échantillonnée de la transformée cosinus. Elle permet de convertir un bloc de 8x8 pixels en un bloc de 8x8 coefficients par une multiplication par une fraction.

Ce procédé présente néanmoins un avantage notable en ce qu'il convertit les données sources en une forme qui facilite la compression.

En ce qui concerne les images couleur, les signaux Y, Cr et Cb sont assemblés en réseaux séparés de 8x8 pixels et traités séparément.

En raison du nombre généralement important de coefficients qui ont une valeur nulle ou proche de zéro, ceux-ci ne sont pas transmis afin de permettre un meilleur taux de compression, sans pour autant entraîner une perte importante de qualité.

Si l'on désire accroître la compression, il est possible de réduire la longueur des mots des coefficients. Néanmoins, cette opération entraîne une baisse de précision des coefficients, ce qui provoque une perte d'information lors des traitements. L'enjeu consiste alors à placer ces pertes de manière judicieuse afin qu'elles soient le moins visibles possible, ce qui suppose de connaître les caractéristiques de l'œil humain.



**Figure6.** Organigramme de MPEG

Si nous prenons une image d'une vidéo, le CODEC regroupe les pixels par lots de 2 lignes et découpera chaque lot tous les 4 pixels à fin d'aboutir à un lot de 8 pixels comme suit :


Tous les capteurs actuels codent la chroma (4 :4 :4, 4 :2 :2, 4 :2 :0) et gardent l'intégralité de la Luma.

Si nous sommes sur le 4 :4 :4 désignant le format le plus lourd en débit et en espace de stockage, nous conserverons chaque valeur de chroma. En effet, le format 4 :4 :4 conserve chaque valeur de chroma et fut dédié à la post-production lourde et précise. Avec une telle précision sur chaque pixel, nous obtiendrons un bon rendu par contre dans le cas d'une diffusion en directe cette technique ne sera pas adéquate.

Passons sur le 4 :2 :2 qui est dédié à l'usage post-production légère et sachant que

- Conserver toujours l'intégralité de l'information de la Luma.
- Regrouper la chroma des pixels 2 à 2 (si on leur rend leurs valeurs de luma ils seront différents de couleurs même si on a perdu un peu en nuance de couleur mais l'œil humain en sera indifférent et la qualité de l'image restera acceptable) comme suit :


Passons sur le 4 :2 :0

- Comme en 4 :2 :2 regrouper les pixels en 2 par 2 sur chaque ligne en calculant la moyenne de chaque 2 pixels comme suit :


- Regrouper chaque lot de 4 pixels entre les 2 lignes ensemble afin de former qu'un seul gros pixel de même couleur regroupant la moyenne des moyennes des 4 pixels qui constituaient cette partie de l'image comme suit :


Notons que :

- **Malgré qu'à la fin nous aboutirons à une chromométrie beaucoup moins précise ; sur une image animée, il est quasiment impossible de percevoir une différence entre 4 :4 :4**

et 4 :2 :2 et il est impossible de zoomer une 4 :2 :0 pour apercevoir sa dégradation par rapport à 4 :4 :4.

→ 4 :4 :4 et 4 :2 :2 (pèse 30% de moins que le 4 :4 :4) sont réservées à des besoins de post-production importants et se contenter de 4 :2 :0 (pèse 50% de moins que le 4 :4 :4).

Si l'on considère que le signal d'entrée est à la norme 4 :2 :2 séries en 10 bits, il est nécessaire de réduire la résolution à 8 bits car le MPEG n'accepte pas de résolution supérieure. Le passage à la norme 4 :2 :0 entraîne une légère mais irréversible perte d'information, couplée à une réduction du débit. (Il est indispensable que le format d'entrée puisse être converti en blocs de 8x8 pixels).

La DCT transforme l'information d'image dans le domaine fréquentiel puis les coefficients sont tronqués et pondérés, ce qui correspond à une première compression.

Les coefficients sont ensuite scrutés en zig-zag pour accroître la probabilité de commencer par les coefficients les plus significatifs. Un code de fin de bloc (EOB = End of Block) est généré après le dernier coefficient non nul.

Généralement, les coefficients les plus significatifs de la DCT se trouvent dans le coin supérieur gauche de la matrice qui contient de nombreuses valeurs tronquées à zéro. Afin d'obtenir une transmission efficace, il semble judicieux d'émettre d'abord tous les coefficients non nuls puis un code qui indique que tous les autres ont une valeur égale à zéro.

La phase du codage entropique est en ordre, des termes nuls apparaissent toujours au milieu des coefficients non nuls. Aussi, le codage RLC (run length coding) a été mis au point afin de gérer plus efficacement les coefficients. Dans le cas de suites de valeurs identiques, comme des zéros, plutôt que d'émettre une suite de bits nuls, ce codage indique simplement le nombre de zéros.

De plus, étant donné que certaines valeurs se rencontrent plus fréquemment que d'autres, une compression à longueur variable (VLC), basée sur ces informations statistiques, peut être proposée. Dans ce cas, les valeurs les plus fréquentes reçoivent des codes plus courts que les valeurs plus rares.

## 6.2 Le codage temporel

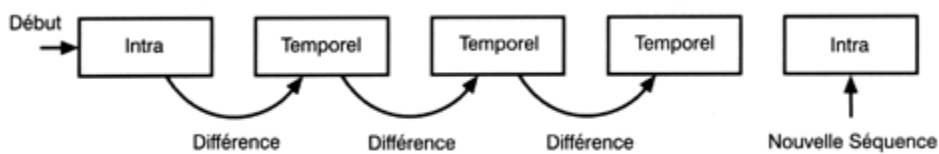
La redondance temporelle peut être exploitée par inter-codage, en transmettant uniquement les différences entre images. La différence est une image à part entière qui peut être compressée en tant que telle par le codeur spatial.

Obtention d'une différence d'image.

Le décodeur inverse ensuite le codage spatial et ajoute l'image de différence à l'image précédente pour obtenir l'image suivante. Ce système simple présente néanmoins quelques inconvénients.

- En raison de la transmission des seules différences, le décodage ne peut pas commencer dès le début de la transmission, ce qui peut poser problème lors d'une commutation d'un flux de données vers un autre (exemple : changement de canal).
- Si une partie des données de différence est incorrecte, l'erreur persiste indéfiniment dans l'image.

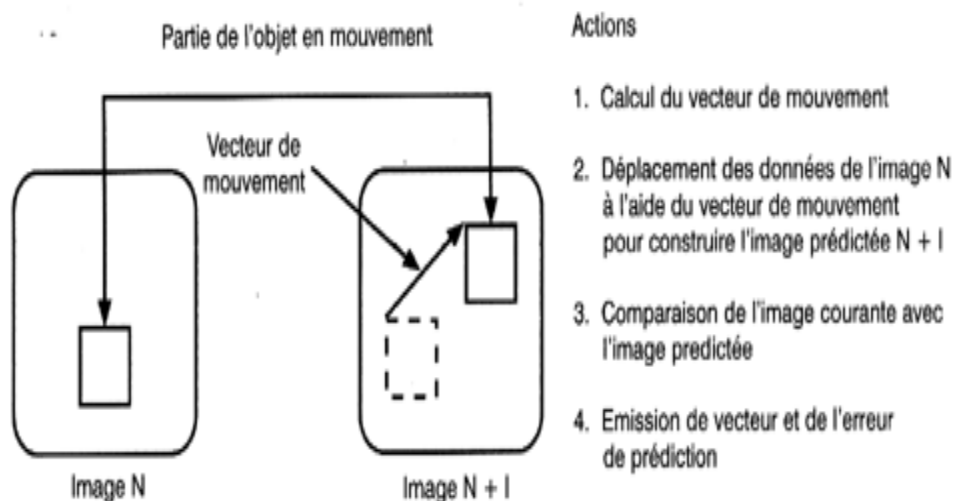
Pour résoudre ces problèmes, la solution d'un système couplant des différences et des images complètes transmises périodiquement (images intra-codées) a été proposée.



### 6.2.1 La compensation de mouvement

L'estimation de mouvement est effectuée sur des macro-blocks Y r-y b-y (une luminance et une chrominance séparées).

Le mouvement réduit la similitude entre deux images et augmente la quantité de données nécessaires à la création de l'image de différence. Aussi, une compensation de mouvement a été mise au point pour tenter d'accroître cette similitude.



Lorsqu'un objet se déplace sur un écran de télévision, il apparaît à un endroit différent mais ne change généralement pas beaucoup d'aspect. Lors du codage, il est ainsi possible de calculer le vecteur de ce mouvement. Ce vecteur est ensuite utilisé par le décodeur pour décaler une partie de l'image précédente vers l'emplacement approprié dans la nouvelle image (La zone de l'image qui est déplacée est appelée Macro-block ).

Afin d'avoir le meilleur taux de compression, les vecteurs de mouvement sont transmis différemment. Par conséquent, si un objet se déplace à vitesse constante, les vecteurs ne changent pas et la différence de vecteur est nulle.

D'autre part, les vecteurs de mouvement sont associés à des macro-blocks et non à des objets réels de l'image, ce qui pose problème lorsque seule une partie du macro-block est amenée à se déplacer. En effet, si le déplacement de la partie mobile est compensé par la transmission d'un vecteur, la partie fixe sera déplacée indûment et devra être corrigée par des données de différence. A l'inverse, si aucun vecteur de mouvement n'est transmis, la partie fixe sera correcte mais il faudra des données de différence pour corriger la partie mobile.

Aussi, afin de répondre convenablement à cette situation, il est indispensable que le compresseur essaye les deux méthodes, en vue de conserver la moins exigeante en données de différence.

### **6.2.2 Le codage bi-directionnel**

Lorsqu'un objet se déplace, il masque progressivement le fond dans la direction du mouvement et le révèle après son passage, ce qui suppose qu'aucune information sur la zone anciennement masquée ne soit disponible dans une image antérieure à celle sur laquelle elle apparaît. Pour résoudre ce problème, le MPEG utilise un codage bi-directionnel qui permet de créer à posteriori des images « antérieures » contenant certaines informations invisibles à un moment donné.

Sur la base d'un macro-block, une image codée bidirectionnellement peut ainsi obtenir des données compensées en mouvement de la part d'une image antérieure, d'une image ultérieure voire faire une moyenne des deux ensembles de données.

- **Les images I, P et B**

En MPEG, trois différents types d'images sont nécessaires pour effectuer le codage différentiel et le codage bidirectionnel avec un minimum d'erreurs de propagation [Richardson, 2010].

- Les images I sont intra-codées (elles ne contiennent pas de vecteur de mouvement et nous cherchons la corrélation entre les pixels sur l'axe spatial → codage intra-image), ce qui permet d'effectuer leur décodage sans apport d'informations extérieures. L'image I garantit une décompression de bonne qualité pour une vidéo.

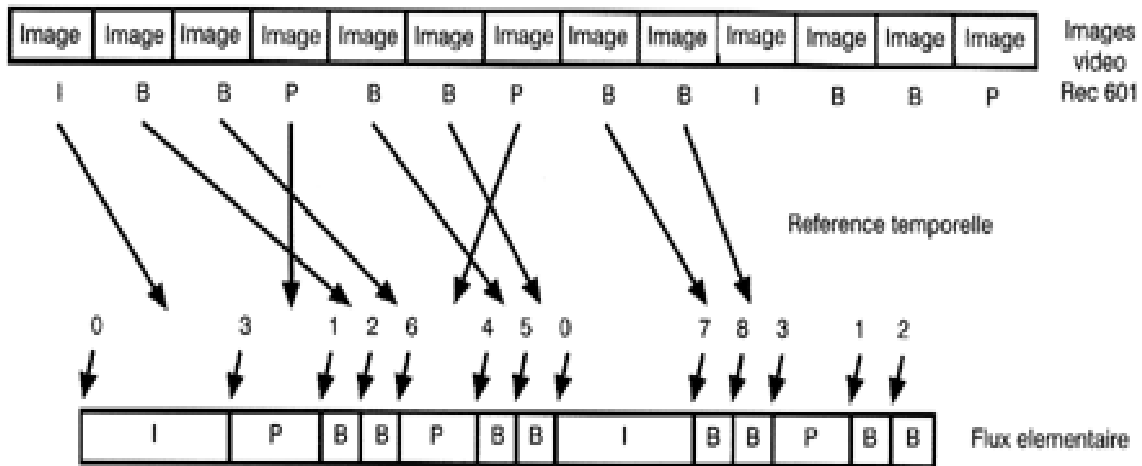
Toutefois, cette caractéristique pose un problème de poids, ce qui explique que leur transmission ne soit réalisée que si elle est indispensable. Elles autorisent la commutation de voies et bloquent la propagation des erreurs.

- Les images P sont déduites d'une image antérieure qui peut être de type I ou P. Les données d'une image P sont constituées de vecteurs décrivant où chaque macro-block doit être pris dans l'image précédente ainsi que de coefficients non transformés qui décrivent la correction ou les données de différence à ajouter à ce macro-block (taille deux fois moindre que pour une image I). Les images P sont susceptibles aux erreurs vu la dépendance complexes aux autres images.

- Les images B sont redictées bidirectionnellement (donc nous aboutirons à une précision de compression élevée) à partir d'images antérieures comme postérieures (elles dépendent des images antérieures ainsi que des futures images), ce qui explique leur poids quatre fois plus réduit que pour une image I.

Les images B recherchent les images I et P, précédentes et futures, les plus ressemblantes pour définir les changements ainsi que les vecteurs de mouvement.

Le GPO (Group of Pictures) est une façon de combiner ces trois types d'image de manière à obtenir le meilleur taux de compression, tout en permettant une commutation en cours de diffusion. D'une longueur d'environ 12 à 15 images, il place une succession d'images P et B entre deux images I. Toutefois, afin que des images B puissent être utilisées afin de construire des images ultérieures dans le cadre d'un codage bi-directionnel, il est indispensable qu'elles soient extraites de la séquence et temporairement sauvegardées.



Ce schéma montre que les données des images P sont émises avant celles des images B qui les précèdent et que les dernières images du GPO ne peuvent être transmises qu'après la première image I du GPO suivant, en raison du fonctionnement du codage bi-directionnel.

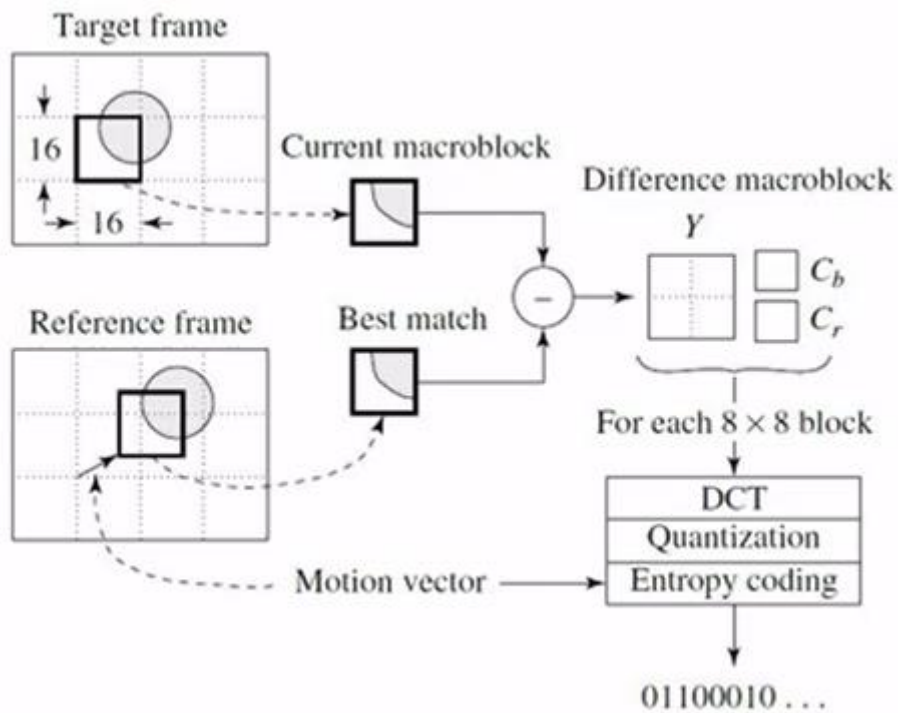
Donc si nous lisons la vidéo nous aurons le séquençement suivant : I B B P B B P B B I B B P Les image B apparaitront avant les images P mais durant l'encodage ou la transmission le séquençement sera le suivant : I P B B P B B I B B. Nous coderons les images I et P en premier après nous pourrons transmettre l'image B du moment que je connais I et P.

Aussi, afin de replacer convenablement les images, une référence temporelle est incluse dans chacune d'entre-elles. Cependant, ce procédé nécessite la présence d'une mémoire dans le décodeur et génère un certain retard d'autant plus important que le taux de compression est élevé.

- **Exemple illustratif**

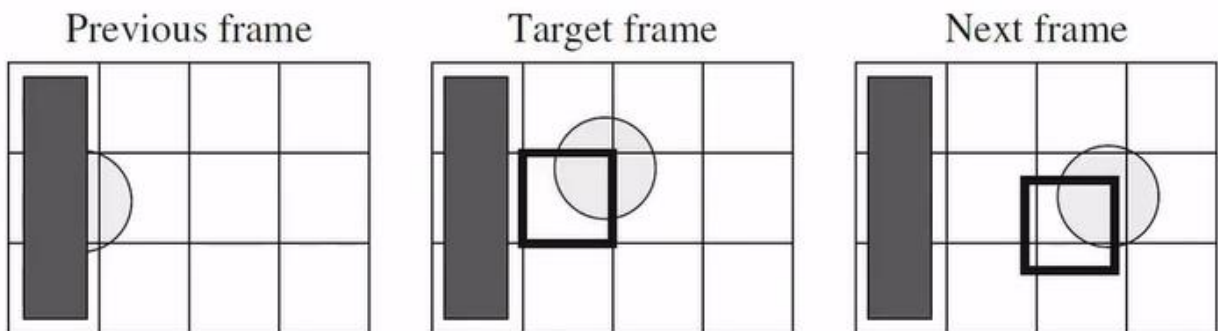
Les images I sont décomposées en macro-block s et seront traitées entant qu'images JPEG (codage intra-image).

A présent les image P désigneront de sauvegarder ou de transmettre la différence entre 2 blocks comme suit :

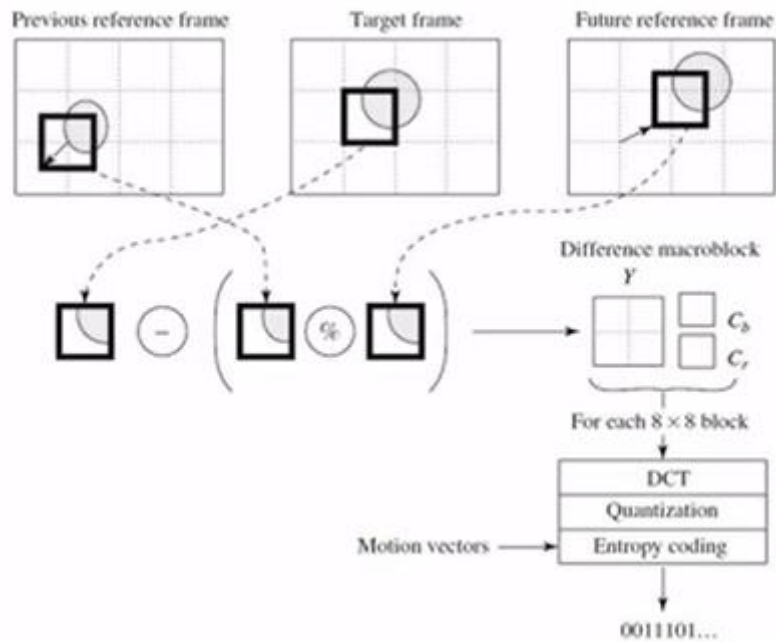


**Figure7.** Codage des images P

Un problème pourra apparaitre :



Supposons que l'objet en mouvement ne soit pas apparu dans l'image I précédente et qu'elle apparait directement au niveau de l'image P et donc nous ne pourrons pas calculer son vecteur de mouvement. MPEG a résolu ce problème par l'intégration des images B bidirectionnelles comme suit :



**Figure8.** Codage des images B

Le principe donc consiste à ne pas considérer uniquement l'image précédente mais à la fois l'image précédente et suivante (si l'objet apparaît dans l'image suivante nous prendrons l'information de cette image et si l'objet apparaît de l'image précédente nous prendrons l'information de l'image précédente). Evidemment, nous sauvegarderons ou transmigrerons la différence. Sur la figure précédente l'objet apparaît dans l'image précédente ainsi que suivante donc nous réaliserons une interpolation entre les 2 images mais si l'objet apparaissait sur uniquement l'image suivante, nous prendrions l'information uniquement de l'image suivante.

- **Mise au point sur les images I B P**

L'image P effectue une compression temporelle via le temps en prédictant une future valeur à partir d'une valeur antécédente. Ex : sur une image P la valeur d'un pixel est égale (par prédiction) à la valeur du pixel sur l'image I + une certaine valeur.

Dans la norme H.264, les images P sont prédites à partir de multiples images I ou B. donc pour décompresser une image P en cours on doit décompresser avant ceci les images I ou B qu'elle en dépende.

Notons que dans MPEG1 l'image en cours ne dépend que d'une seule image précédente (un retour en arrière uniquement par une seule position).

Notons que dans l'image P on ne stocke pas tous les blocks mais uniquement celui qu'on va prédire et qui a été prouvé qu'il a changé de position ou de mouvement.

Notons qu'il est souhaitable d'utiliser autant d'image P que possible vu que nous utilisons des blocks déjà existants auparavant et que nous codons que les nouvelles données apparues.

Maintenant nous nous trouvons face au cas où une image P dépend par exemple d'une image P précédente et qu'un nouvel objet apparaît dedans → nous devons la coder avec l'image suivante aussi → intégration de l'image B pour prédire à partir des images I et P précédentes et futures.

### 6.2.3 Le codage de l'audio

· Codage en sous-bandes:

Afin de prendre en compte les caractéristiques de l'audition humaine, des compresseurs à bandes séparées ont été mises en place. Ils découpent le spectre audio en bandes de même largeur qui se recouvrent en partie. (En MPEG, 32 bandes découpées chacune en 12 échantillons sont distinguées).

Dans chaque bande de fréquence, l'entrée audio est amplifiée au maximum avant la transmission puis chaque niveau est ensuite ramené à sa valeur initiale, ce qui permet de réduire le bruit introduit par la transmission. Toutefois, il apparaît qu'un bruit plus important peut être toléré dans certaines bandes sans passer un seuil d'audition, ce qui autorise de réduire la taille des mots d'échantillonnage avant la compression par un processus de quantification.

La longueur de mots est alors transmise comme un code d'affectation de bits, afin de permettre au décodeur de désérialiser convenablement le flux de bits.

· Couche 1 du MPEG:

Après le mot de synchronisation qui est détecté par le générateur de temps et l'en-tête, 32 codes d'affectation de bits de 4 bits chacun décrivent la longueur des mots des échantillons dans chaque sous-bande. Viennent ensuite les 32 facteurs d'échelle utilisés par la compression dans chaque bande puis les données audios de chaque bande.

Le décodeur applique une re-quantification inverse puis une multiplication par l'inverse du facteur de compression, de façon à ramener le niveau de chaque bande à sa bonne valeur. Les 32 bandes sont ensuite rassemblées dans un filtre de recombinaison pour rétablir la sortie audio.

· Couche 2 du MPEG:

Afin d'améliorer la précision de la résolution de fréquence, l'excursion temporelle de la transformée est accrue (blocs constitués de 1152 échantillons). Contrairement à la couche 1 du MPEG, tous les facteurs d'échelle ne sont pas transmis dans la mesure où, ils peuvent présenter un degré de redondance non négligeable. (Aussi, sur les programmes fixes, seul un facteur d'échelle sur trois est transmis).

· Couche 3 du MPEG:

Alors que les couches 1 et 2 du MPEG sont basées sur des filtres séparateurs de bande dans lesquels le signal est toujours représenté comme une forme d'onde, la couche 3 utilise un codage à transformée comme celui utilisé en vidéo. Cette dernière est utilisée lorsque des facteurs de compression élevés sont nécessaires.

Une transformée cosinus discrets à 384 coefficients de sortie par bloc est appliquée soit par un traitement direct des échantillons d'entrée, soit à l'aide d'un codeur multi-niveaux qui permet d'utiliser une transformée hybride incorporant le filtrage 32 bandes des couches 1 et 2. Dans ce cas, les 32 sous-bandes du filtre QMF (Quadrature Mirror Filter) sont ensuite traitées par une Transformée Cosinus Discrète Modifiée (MDCT) à 32 bandes pour obtenir les 384 coefficients. Pour obtenir le facteur de compression le plus élevé, la quantification non uniforme des coefficients selon le codage de Huffman attribue les mots les plus courts aux valeurs de code les plus fréquentes [Sayood, 2017 ; Bovik, 2005].

## **7. Les conteneurs vidéo**

Chaque conteneur accepte différents codecs audios et vidéos et permet la présence d'une ou plusieurs pistes audios, de sous-titres intégrés ou non, de chapitres, et éventuellement d'autres informations [Bovik, 2005]. Il permet ainsi de mettre un film complet dans un seul fichier, de manière à ce que toutes les informations soient lisibles par la plupart des lecteurs multimédias. Afin de clarifier le vocabulaire, il est important de souligner que certains formats de fichiers désignent à la fois le conteneur et le codec, ce qui peut parfois prêter à confusion.

Les principaux conteneurs sont :

- AVI : il peut contenir une piste vidéo au codec MPEG1, MPEG2, DivX, XviD ainsi qu'une piste audio au codec MP3 ou WMA.
- BivX : variante de l'AVI qui permet d'intégrer une seconde bande-son.
- Ogg : il contient du son ogg vorbis et/ou de la vidéo ogg théora.
- Ogm : hack de l'AVI qui permet d'intégrer des pistes sons ogg vorbis ainsi que des sous-titres

- Matroska (MKV) : il permet théoriquement d'importer n'importe quels codecs vidéo et audio ainsi que plusieurs canaux de sous-titres, des menus style DVD voire des pièces jointes. Il permet également un chapitrage et sa structure XML facilite la recherche dans les méta-tags.
- Divx6 : il autorise l'incorporation de menus identiques à ceux présents dans les DVD vidéo mais n'est pas libre.
- MP4 : conteneur officiel défini dans le standard MPEG4.
- NUT : conteneur libre.
- WMV : conteneur propriétaire Windows qui contient exclusivement des pistes vidéo au codec Microsoft Windows Media.

Matroska (du mot russe matriochka = poupée russe) : ce conteneur Open Source est dérivé du projet MCF mais en diffère toutefois car il utilise EBML (extensible binary meta language), un format binaire dérivé d'XML. Ce format présente l'avantage de permettre aux développeurs de rajouter aisément des extensions au format, tout en conservant une compatibilité avec les anciens logiciels de lecture (ajout de modules). La taille des fichiers est équivalente au DIVX.

Il permet une diffusion à travers internet (HTTP et RTP), une recherche rapide dans le fichier, une résistance aux erreurs, la création de menus, un chapitrage, un choix entre plusieurs sous-titres ainsi qu'un choix entre plusieurs pistes audio. Les codecs audios supportés incluent Ogg Vorbis, AAC, MP3, et AC3. Les codecs vidéo supportés incluent le DivX, le XviD, le RV10, le H264 ainsi que l'AAC.

## **8. Aperçu sur le processus du décodage**

Au niveau du décodeur, le flux de bits est désérialisé et le codage de l'entropie est inversé pour reproduire les coefficients pondérés. La pondération inverse est ensuite appliquée et les coefficients sont replacés dans les matrices en fonction de la scrutation en zig-zag afin de recréer les matrices DCT. A la suite de la transformée inverse, les blocs de 8x8 pixels sont alors recréés.

Pour obtenir un signal de sortie d'écran balayé, les blocs sont stockés dans une RAM lue ligne par ligne. La conversion de données 4 :2 :0 en données 4 :2 :2 nécessite quant à elle un recours à une interpolation verticale.

Le MPEG conduit à des systèmes fortement asymétriques : le processus de compression est beaucoup plus complexe que le processus de décompression. De plus, il faut une puissance de calcul bien supérieure pour la compression que pour la décompression.

Cette caractéristique ne pose néanmoins pas de problèmes lorsqu'il s'agit de diffuser des images car la compression s'effectue alors en un endroit unique.

Toutefois, en dépit de ses nombreux avantages, le MPEG n'est pas utilisé tout au long des chaînes de production de vidéos professionnelles, du fait de la nécessité de compresser et de décompresser à chaque maillon de la chaîne.

- Le système MPEG n'a pas été conçu pour faire du montage à l'image près, ce qui est un des prérequis majeurs pour faire de la postproduction.
- Des générations successives, entrecoupées de traitements (effets), peuvent induire une perte de qualité qui s'avérera rapidement inacceptable.

## Référence

- Bracewell, R. N. (2000). *The Fourier Transform and Its Applications* (3rd ed.). McGraw-Hill.
- Burger, W. & Burge, M. J. (2016). *Digital Image Processing: An Algorithmic Introduction Using Java* (2nd ed.). Springer.
- Cover, T. M. & Thomas, J. A. (2006). *Elements of Information Theory* (2nd ed.). Wiley.
- Gonzalez, R. C. & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson.
- Oppenheim, A. V. & Schafer, R. W. (2010). *Discrete-Time Signal Processing* (3rd ed.). Pearson.
- O'Shaughnessy, D. (2000). *Speech Communication: Human and Machine* (2nd ed.). IEEE Press.
- Pennebaker, W.B. & Mitchell, J.L., 1992. *JPEG Still Image Data Compression Standard*. Springer.
- Pratt, W. K. (2007). *Digital Image Processing* (4th ed.). Wiley-Interscience.
- Sayood, K. (2017). *Introduction to Data Compression* (5th ed.). Morgan Kaufmann.
- Skodras, A., Christopoulos, C. & Ebrahimi, T., 2001. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5), pp.36–58.
- Smith, J. O. (2007). *Introduction to Digital Filters with Audio Applications*. W3K Publishing.
- Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer.
- Taubman, D.S. & Marcellin, M.W., 2002. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Springer.
- Bracewell, R. N. (2000). *The Fourier Transform and Its Applications* (3rd ed.). McGraw-Hill.
- Bovik, A. C. (Ed.) (2005). *Handbook of Image and Video Processing* (2nd ed.). Academic Press.
- Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson.
- Oppenheim, A. V., & Schafer, R. W. (2010). *Discrete-Time Signal Processing* (3rd ed.). Pearson.
- Richardson, I. E. G. (2010). *The H.264 Advanced Video Compression Standard* (2nd ed.). Wiley.
- Sayood, K. (2017). *Introduction to Data Compression* (5th ed.). Morgan Kaufmann.
- Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer.

Watkinson, J. (2008). *The MPEG Handbook: MPEG-1, MPEG-2, MPEG-4* (2nd ed.).  
Routledge.