L'imagerie numérique/ Compression

- **Introduction**

  La compression est l'action utilisée pour réduire la taille physique d'un bloc d'information. En compressant des données, on peut placer plus d'informations dans le même espace de stockage, ou utiliser moins de temps pour le transfert à travers un réseau téléinformatique.

Les critères d'évaluation de la compression

Il existe une grande variété de techniques de compression de données. Afin de pouvoir comparer les performances de ces méthodes, différents critères sont nécessaires.

Posons '***n*** ', le nombre de bits de la séquence de donnée originale, et '***m*** ', le nombre de bits résultant de la méthode de compression.

L'imagerie numérique/ Compression

- Le ***ratio*** de compression est défini comme le ratio du nombre de bits de la séquence originale au nombre de bits produits par la méthode de compression "**n/m**".
- Le ***taux*** de compression est défini comme le ratio de la différence entre le nombre de bits original et après compression sur le nombre de bit original, soit "***(n−m)/n***". En prenant n =5 et m =1, on trouve que le taux de compression est 0.8, que l'on note parfois comme 80%.

L'imagerie numérique/ Compression

- l'erreur quadratique moyenne (MSE) est souvent considérée comme une bonne mesure de l'exactitude d'une évaluation de l'image. Ce critère est utilisé pour déterminer le rapport de ressemblance, en comparant l'image dégradée Î, de taille M*N, à l'originale I.
  Ce critère est défini comme suit

$$EQM = \frac{1}{M * N} \sum_{i}^{N} \sum_{j}^{M} \left| I_{i,j} - \hat{I}_{i,j} \right|^{2}$$

L'imagerie numérique/ Compression

- Rapport signal sur bruit en pic (PSNR) une autre mesure de la qualité de l'image approchée est le rapport signal sur bruit en pic (PSNR) qui est inversement proportionnel à la MSE, son unité est le décibel (dB) et est défini par la relation suivante

$$PSNR_{dB} = 10log_{10}\left(\frac{L^2}{EQM}\right)$$

Pour une image à niveau de gris, L désigne la luminance maximale possible (255 est la valeur maximale d'un pixel pour une image codée sur 8bits/pixel). le PSNR est l'évaluation la plus couramment utilisée.

L'imagerie numérique/ Compression

- **Les techniques de base de la compression**

La compression consiste à réduire le volume de données nécessaire à la description des informations. L'exploitation de la redondance permet de développer des méthodes où la quantité de données nécessaires pour représenter l'information est réduite.

Il existe deux catégories bien distinctes de compression : la compression sans perte et la compression avec pertes. On présente ci dessous quelques méthodes de compression appartenant à ces deux catégories.

L'imagerie numérique/ Compression sans pertes

Dans le contexte de la compression sans perte, la méthode prend en entrée une série de bits ***X*** qu'elle transforme en une nouvelle série de bits ***Y*** plus courte que ***X***. La série de bits ***Y*** est transmise ou stockée pour usage ultérieur. Lorsqu'on veut récupérer les données, on prend ***Y*** et on applique la méthode de compression inverse -la méthode de décompression- pour récupérer ***X*** intact (une restitution exacte).

Ces méthodes ont été établies dans le cadre de la théorie de l'information. Elles se basent sur des méthodes de codage qui transforme la source dans une forme alternative, meilleure pour la transmission ou pour le stockage.

L'imagerie numérique/ Compression sans pertes

A. Le codage de Huffman

développé par David A. Huffman en1952, est un algorithme de compression des données basé sur les fréquences d'apparition des caractères apparaissant dans le document initial.

Le tableau suivant présente le codage ASCII de la séquence "go go gophers".

Caractère	ASCII	Binaire
G	103	1100111
O	111	1101111
P	112	1110000
H	104	1101000
E	101	1100101
R	114	1110010
S	115	1110011
Espace	32	1000000

L'imagerie numérique/ Compression sans pertes

Caractère	ASCII	Binaire
G	103	1100111
O	111	1101111
P	112	1110000
H	104	1101000
E	101	1100101
R	114	1110010
S	115	1110011
Espace	32	1000000

La suite de 104 bits correspondante à la séquence à comprimer est:

1100111 1101111 1100000 1100111 1101111 1000000 1100111 1101111 1110000 1101000 1100101 1110010 1110011

L'imagerie numérique/ Compression sans pertes

- On verra comment on peut appliquer un codage de Huffman en utilisant un arbre binaire qui contient les caractères dans ses feuilles et la séquence des bits (utilisés pour le codage des caractères) dans ses branches.

- L'arbre binaire de Huffman est la structure de données qui va nous permettre d'attribuer à chaque symbole une représentation binaire optimale. Afin de construire l'arbre, on utilise la table de fréquences d'apparition qu'on appelle T et on applique l'algorithme suivant :

L'imagerie numérique/ Compression sans pertes

Données :
- T : la table de fréquence
- Q : Une file d'attente de noeuds de l'arbre binaire. Chaque feuille est étiquetée avec un symbole et son nombre d'occurrences. Chaque noeud interne est étiqueté avec la somme des occurrences des feuilles de sa sous arborescence.
- o : Une fonction qui à chaque noeud de l'arbre associe une valeur. Si le noeud est une feuille alors o renvoie le nombre d'occurrences du symbole, autrement o renvoie la somme des occurences des feuilles de la sous-arborescence du noeud.

Résultat :
- A : L'arbre binaire résultant

```
begin
    Initialisation de Q tel que :
    Q contienne les feuilles représentant les symboles de la table T

    tant que (Q non vide) faire
        Créer un nouveau noeud z dans A tel que :
        gauche(z) = x = extraire-min (Q)
        droite(z) = y = extraire-min (Q)
        o(z) = o(x) + o(y)
        Insérer (z,Q)

    fin
end
```

L'imagerie numérique/ Compression sans pertes

De façon informelle, on utilise une file d'attente Q dans la quelle on place les nœuds correspondants au couple [symbole : nombre d'occurrences du symbole] de tous les symboles. Ensuite on extrait de la file d'attente les 2 nœuds ayant la valeur minimale puis on crée un nouveau nœud dans l'arbre de Huffman ayant pour fils les 2 nœuds sélectionnés, on rajoute ensuite le nœud nouvellement crée dans la file d'attente, et on réitère jusqu'à ce que la file soit vide.

L'imagerie numérique/ Compression sans pertes

L'algorithme de Huffman suppose la construction d'un seul arbre appartenant à un foret. Au commencement tous les arbres ont un seul nœud qui contient un caractère et sa fréquence d'apparition. On combine les arbres par la sélection de deux arbres qui seront utilisés pour la construction d'un nouvel arbre. Ainsi se réduit le nombre d'arbres de foret avec 1 à chaque pas de l'algorithme. Les étapes de l'algorithme sont les suivantes :

1. On commence avec un foret. Chaque arbre de ce foret a un seul nœud qui contient un caractère et une fréquence d'apparition. On combine les arbres par la sélection de deux arbres qui seront utilisés pour la construction d'un nouvel arbre. Ainsi se réduit le nombre des arbres de foret avec 1 à chaque pas de l'algorithme.

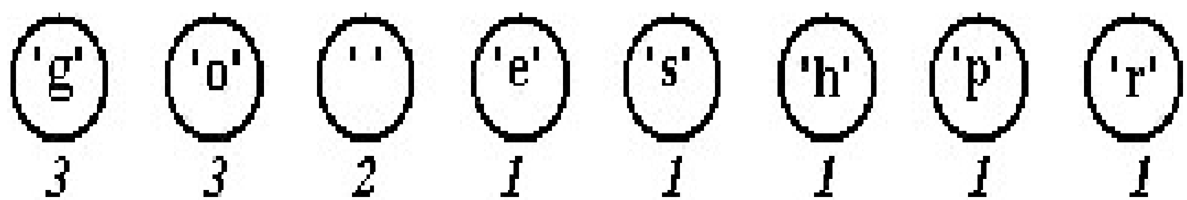
L'imagerie numérique/ Compression sans pertes

2. On répète l'étape antérieure jusqu'à l'obtention d'un seul arbre. On choisi deux arbres avec les plus petites fréquences d'apparition et on les note par T1 et T2. On fait la construction d'un nouvel arbre dont la fréquence d'apparition sera la somme de deux fréquences d'apparition de deux arbres parents T1 + T2 et dont le sous-arbre de gauche est celui de fréquence T1 et le sous-arbre de droite est celui de fréquence T2.

3. Le seul arbre resté après l'étape antérieure est un arbre de codage optimal.

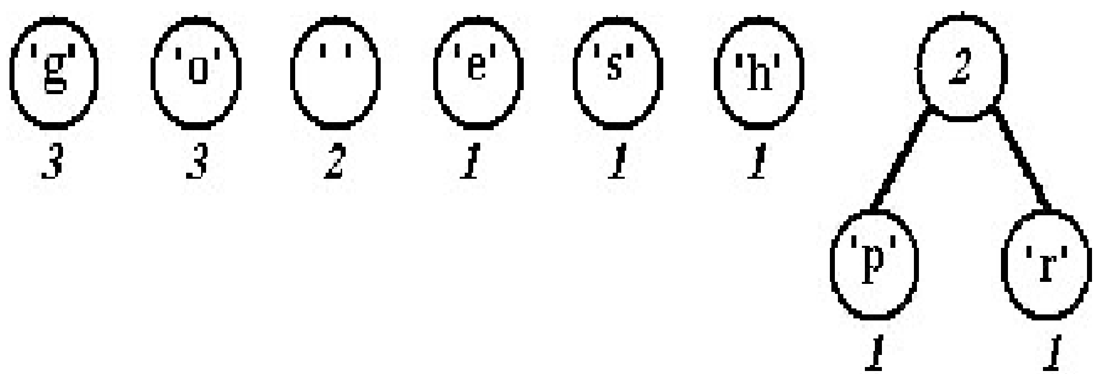
L'imagerie numérique/ Compression sans pertes

En prenant de nouveau l'exemple de la séquence "go go gophers" on a au commencement le foret présenté dans la figure suivante.



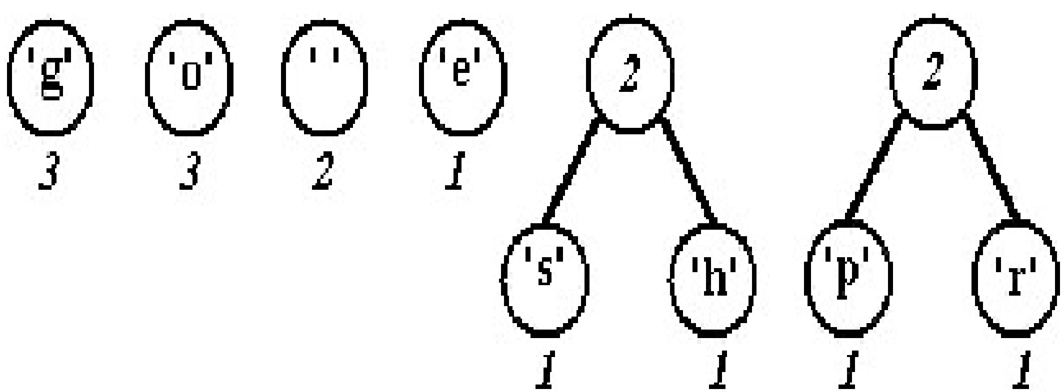
Les fréquences d'apparition de chaque nœud sont proportionnelles aux valeurs indiquées au-dessous de chacun.

On choisit deux nœuds à fréquences d'apparition minimales. On fait la construction d'un nouvel arbre dont la racine est pondérée par la somme des fréquences d'apparition de deux arbres sélectionnés. On a obtenu ainsi le foret, contenant sept arbres



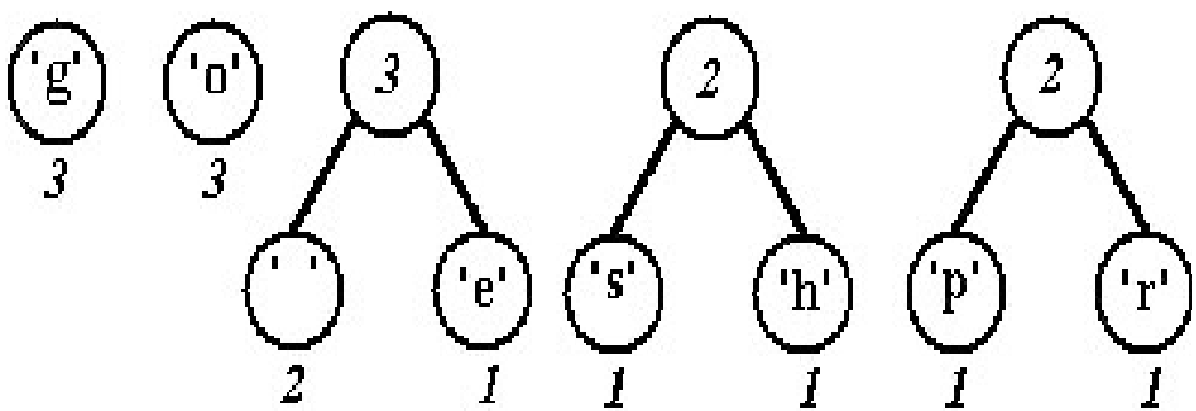
L'imagerie numérique/ Compression sans pertes

En sélectionnant de nouveau deux arbres minimaux on obtient un nouvel arbre à fréquence d'apparition proportionnelle à 2 et le foret présenté dans la figure suivante



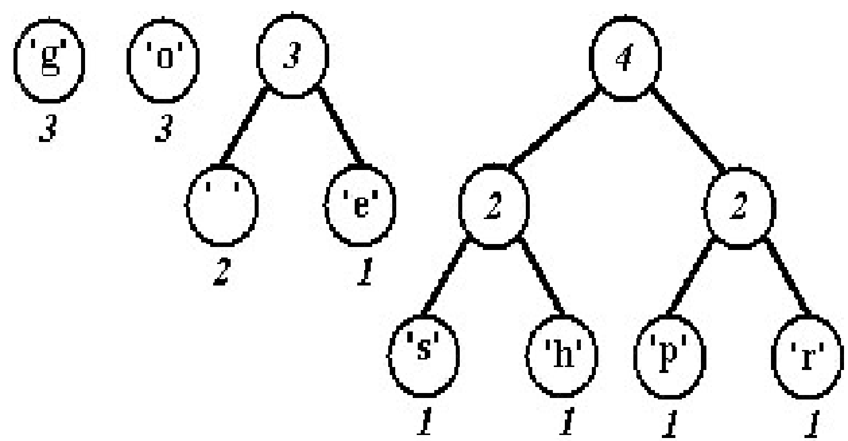
L'imagerie numérique/ Compression sans pertes

Il faut choisir de nouveau les deux arbres de fréquences minimales. Le nœud à fréquence minimale est celui qui corresponde à la fréquence proportionnelle à 1 (celui qui contienne le caractère 'e'). Il y a trois arbres qui ont la fréquence proportionnelle à 2, on peut choisir n'importe lequel pour trouver la paire de l'arbre correspondent au caractère 'e'. Le nouvel arbre créé aura une fréquence d'apparition proportionnelle à 3.

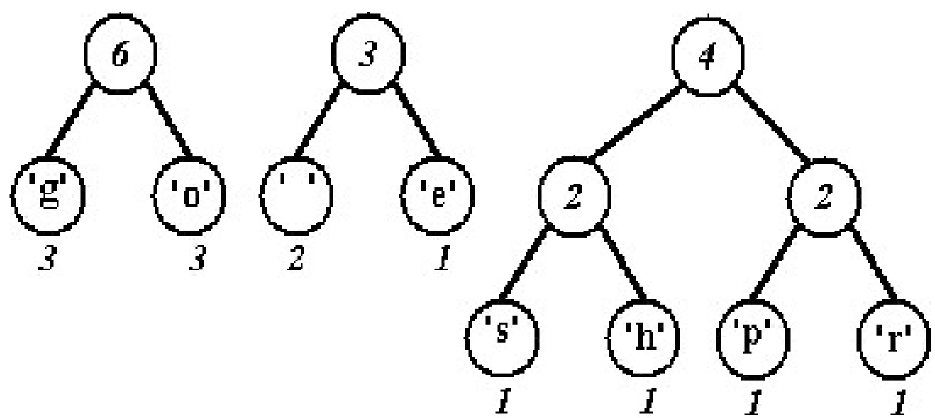


L'imagerie numérique/ Compression sans pertes

Maintenant on a deux arbres à fréquences d'apparition proportionnelles à 2. On utilise ces arbres pour construire un nouvel arbre qui aura la fréquence d'apparition proportionnelle à 4. On a resté quatre arbres, un à fréquence d'apparition proportionnelle à 4 et trois à fréquences d'apparition proportionnelles à 3.

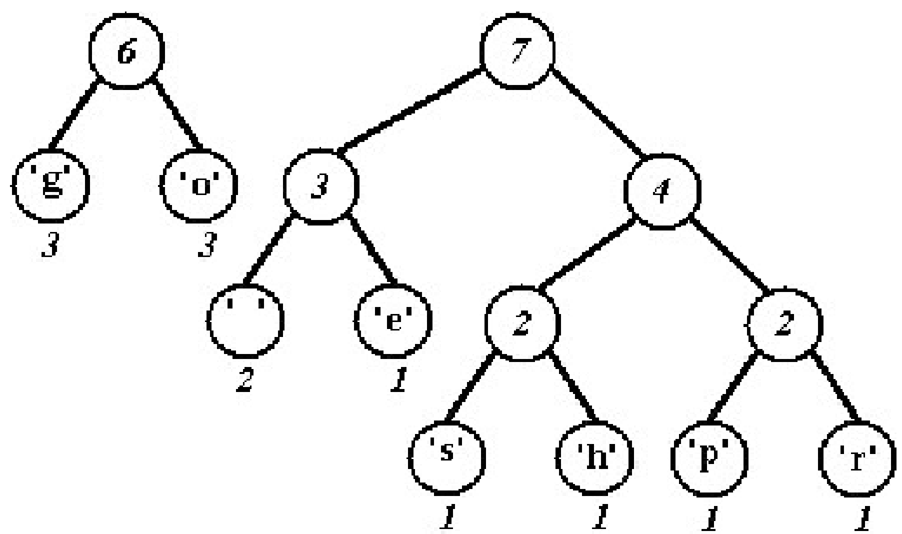


Deux arbres minimaux sont de nouveau associés pour construire un nouvel arbre qui aura une fréquence d'apparition proportionnelle à 6. On a sélectionné les arbres 'g' et 'o'. Le foret obtenu est présentée dans la figure Huff_6. Elle contient trois arbres. Les arbres minimaux ont des fréquences d'apparition proportionnelles à 3 et à 4.

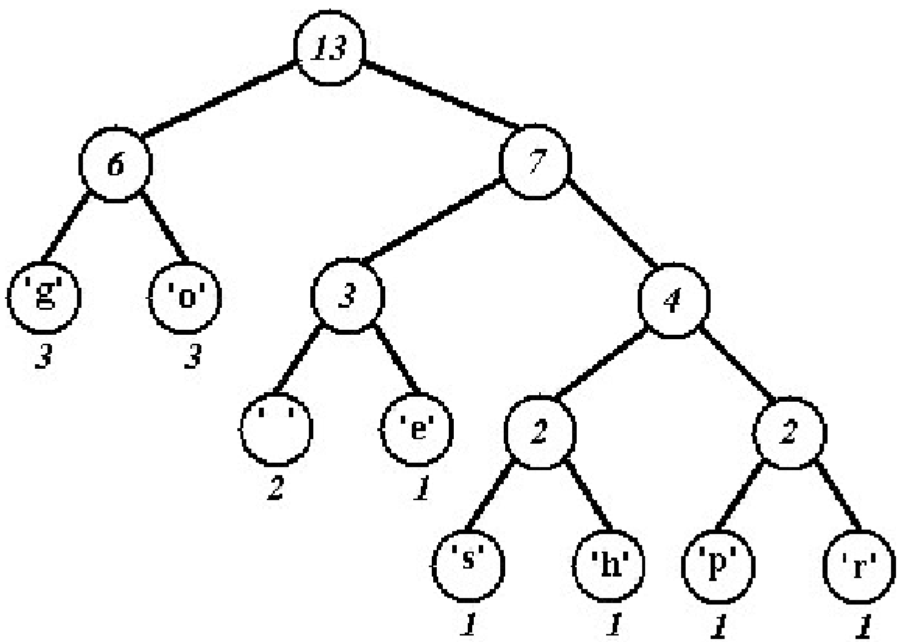


L'imagerie numérique/ *Compression sans pertes*

En les utilisant on peut construire un nouvel arbre qui aura une fréquence d'apparition de 7 et on obtient le foret de deux arbres de la figure suivante.



En fin les deux derniers arbres sont utilisés pour la construction de l'arbre final dont la fréquence d'apparition est proportionnelle à 13.



L'imagerie numérique/ Compression sans pertes

Le codage induit par le dernier arbre est présenté dans le tableau suivant.

Caractère	Binaire
'g'	00
'o'	01
'p'	1110
'h'	1101
'e'	101
'r'	1111
's'	1100
' '	100

En utilisant ce tableau, la séquence "go go gophers" sera codée par la séquence binaire suivante:

00 01 100 00 01 100 00 01 1110 1101 101 1111 1100

On a obtenu une séquence binaire de 37 bits.

L'imagerie numérique/ Compression sans pertes

Le codage de type RLE "run length encoding" :

RLE est l'une des méthodes les plus anciennes, les plus simples et la plus utilisée. Tout son secret consiste à identifier et supprimer des redondances d'informations en les codant sous une forme plus compacte.

Le codage remplace une suite de caractères identiques par le nombre de répétitions. Pour indiquer un tel remplacement on utilise un caractère spécial. Le codage "run-lenght" remplace la suite des caractères identiques X par la séquence:

$$S_c X C_c$$

L'imagerie numérique/ Compression sans pertes

$$S_c X C_c$$

Où :

S_c est un caractère spécial

X est le caractère qui se répète

C_c représente le nombre de répétition.

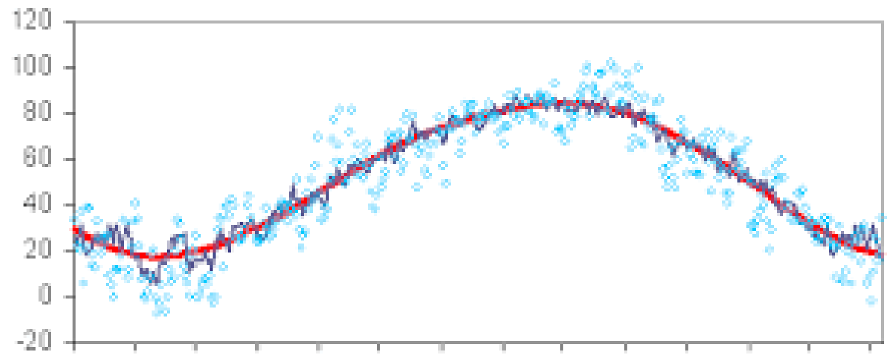
La compression RLE est régie par des règles particulières permettant de compresser lorsque cela est nécessaire et de laisser la chaîne telle quelle lorsque la compression induit un gaspillage. Par conséquent le codage "run-lenght" est efficient seulement si le nombre de répétitions est supérieur à 4

L'imagerie numérique/ Compression avec pertes

Dans la compression avec perte nous ne cherchons pas à coder les signaux de façon exacte, mais nous tenterons d'éliminer le bruit et ne coderons que ce qui est important dans ces signaux. La détermination des composantes du signal qui correspondent au bruit dépend naturellement de la nature du signal considéré et c'est en général un problème qui n'est pas très bien résolu.

L'imagerie numérique/ Compression avec pertes

Les principales méthodes de compression avec perte reposent sur des techniques de décomposition de signal et sur des méthodes de discrétisation (quantization) de scalaires et de vecteurs. Ces méthodes commencent par décomposer le signal d'intérêt en sous-bandes. Typiquement, cette décomposition est faite grâce à une technique d'analyse de fréquence de type transformée de Fourier (et ses descendants) ou plus récemment grâce aux techniques à base d'ondelettes. Une fois la décomposition du signal obtenue, on procédé a une analyse des composantes.



L'imagerie numérique/ Compression avec pertes

Les composantes que l'on associe au bruit sont détruites et les composantes que l'on conserve seront discrétisées avant d'être codées par un algorithme de modélisation statistique. La discrétisation, c'est simplement une réduction de précision, mais une réduction intelligente. Les méthodes de discrétisation permettent de calculer pour un très grand nombre de valeurs distinctes n valeurs qui les représenteront, de façon à minimiser l'erreur moyenne de reconstruction tout en minimisant le nombre de bits requis pour l'encodage de ces n valeurs représentatives.

L'utilisation d'un algorithme avec perte peut conduire à la perte de l'information, mais assure généralement un rapport de compression plus élevé.

L'imagerie numérique/ Compression avec pertes

La plupart des algorithmes de compression avec pertes font appel à la technique de transformation de données qui permet de fournir des informations facilement compressibles. L'idée est qu'en effectuant une transformée linéaire sur les données originales, on obtient un ensemble de coefficients dans l'espace transformé dont les composantes sont moins corrélées entre elles. Dans la plupart des cas, un nombre réduit de coefficients quantifiés et non nuls est suffisant pour récupérer une approximation des données originales avec une faible distorsion. Le codage d'entropie est généralement appliqué aux coefficients afin de réduire encore le taux de données.

L'imagerie numérique/ Compression avec pertes

La famille des transformées orthogonales est fortement recommandée pour ce domaine d'application du moment que la transformée inverse est nécessaire pour la reproduction du signal original et que l'inverse d'une matrice orthogonale est sa transposée.

Supposons que l'on ait un bloc de *N* échantillons (vecteur) d'un processus aléatoire stationnaire $X = \{x_0, x_1, \dots, x_{N-1}\}^T$.

D'après l'hypothèse de stationnarité, ces échantillons ont la même variance et peuvent présenter une corrélation importante. Cette corrélation entraîne une certaine redondance qui est conservée dans les échantillons quantifiés

L'imagerie numérique/ Compression avec pertes

On introduit alors le vecteur $Y = \{y_0, y_1, \dots, y_{N-1}\}^T$ obtenu par transformée orthogonale linéaire T de sorte que $Y = TX$. On quantifiera alors les composantes du vecteur transformé afin d'obtenir le vecteur $\hat{Y} = Q(Y)$. Au décodage, le vecteur $\hat{X}$ sera reconstruit par transformation inverse du vecteur quantifié

$\hat{X} = T^{-1}\hat{Y}$. On impose à la transformation T, la condition d'orthogonalité $T^{-1} = T^T$.

Le choix de la méthode de transformation doit permettre une réduction de la corrélation des données d'entrée. La sélection de la transformation est un facteur important dans la compression des données.

La transformation qui permet d'obtenir un ensemble de coefficients non-corrélés et qui minimise la distorsion est la transformation de Karhunen-Loeve (*TKL*). Cette transformation est formée par les vecteurs propres de la matrice de covariance du signal X et est donc dépendante du signal traité.

L'imagerie numérique/ Compression avec pertes

D'autre part, il n'existe pas d'algorithme rapide pour le calcul de la *TKL* qui nécessite par ailleurs l'évaluation de la matrice d'auto-corrélation pour des signaux qui ne sont pas stationnaires. C'est le motif pour lequel en pratique on utilise des transformées sous optimales, comme par exemple la transformée en cosinus discrète (DCT est utilisé en JPEG) ou la transformée en ondelettes discrète (DWT est utilisé dans JPEG2000 et EZW). Ce deux transformées convergent asymptotiquement vers la transformée de Karhunen-Loève.

L'imagerie numérique/ Compression JPEG

L'algorithme JPEG commence par découper l'image en blocs carrés (généralement de 8×8 pixels). L'étape suivante dans le processus de compression consiste à appliquer une transformée en cosinus discrète (DCT) à chaque bloc 8x8 pixels. C'est l'étape qui nécessite plus de temps et de ressources dans la compression et la décompression, mais la plus importante car elle nous a permis de séparer les basses fréquences et les hautes fréquences présentes dans l'image.

L'imagerie numérique/ Compression JPEG

L'étape DCT lui-même est sans perte. La quantification est l'étape dans laquelle on perd réellement des informations (et donc de la qualité visuelle), mais c'est celle qui fait gagner beaucoup de place. Le but est de garder les basses fréquences, c'est-à-dire celles aux quelles l'œil humain est très sensible et de vouloir annuler les hautes fréquences, c'est-à-dire celles aux quelles l'œil humain est moin sensible.

L'imagerie numérique/ Compression JPEG

A l'étape suivante, JPEG utilise des tables de quantification pour quantifier les coefficients DCT. Chaque coefficient DCT est divisé par sa constante correspondante dans la table de quantification et arrondie au plus proche entier.

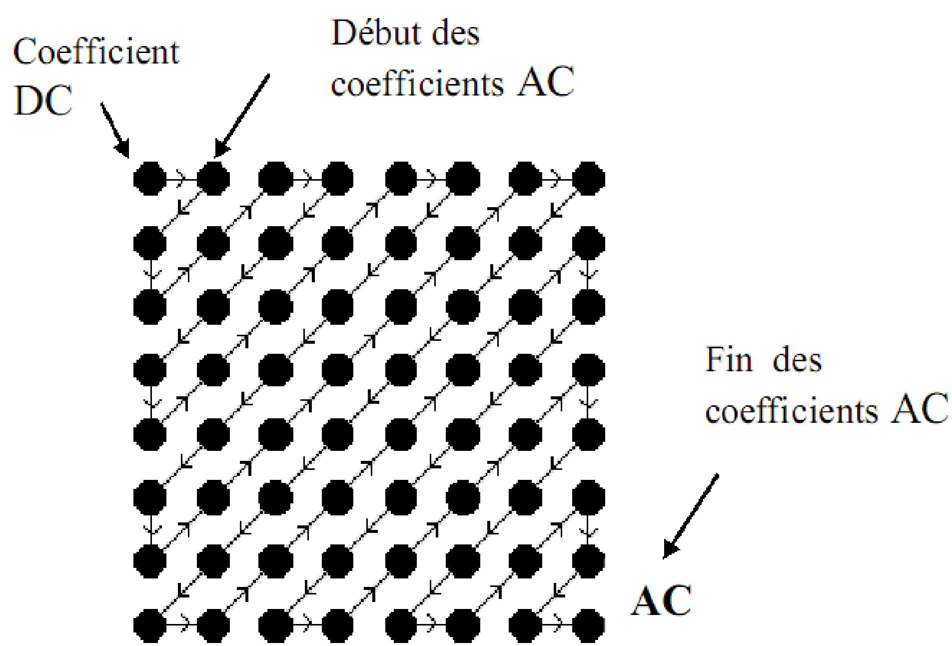
```
table_lum =
```

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

L'imagerie numérique/ Compression JPEG

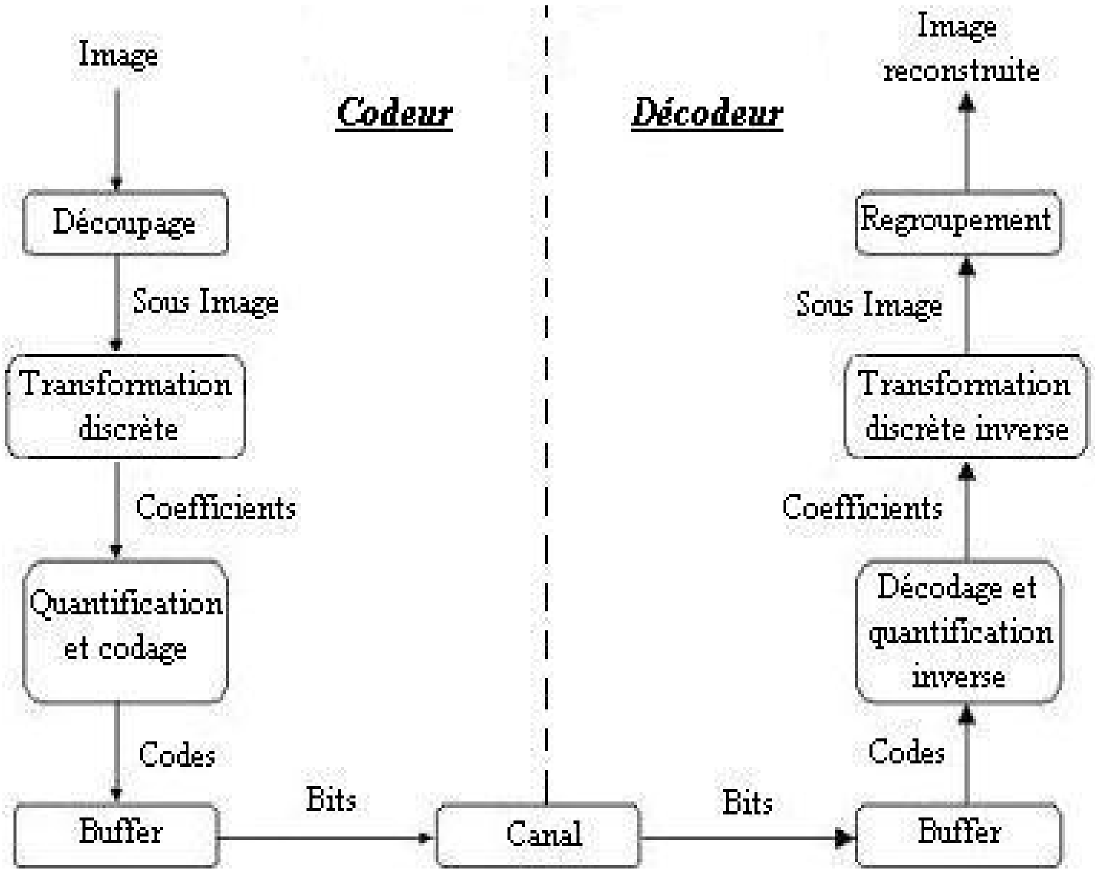
Le résultat de la quantification des coefficients DCT sont parcouru en zigzag avant d'appliquer le codeur entropique (RLE puis HUFFMAN).

Pour pouvoir reconstruire l'image originale à partir de l'image compressée, il faut appliquer le procédé inverse (décompression).



L'imagerie numérique/ Compression JPEG

Le principe du système de codage par transformation JPEG pour les images est représenté dans la figure suivante:



L'imagerie numérique/ JPEG / DCT

DCT est une transformée semblable à la transformée de Fourier rapide (FFT). La DCT type II transforme un ensemble de données d'un domaine spatial en un domaine "fréquentiel" et inversement (IDCT II).

Elle 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.

La DCT II est effectuée sur une matrice carrée $N \times N$ de valeurs de pixels $f(x, y)$ et donne une matrice carrée $N \times N$ de coefficients de fréquence $F(u, v)$. Le temps de calcul requis pour chaque élément dans la DCT II dépend de la taille de la matrice.

L'imagerie numérique/ JPEG / DCT

Vu la difficulté d'appliquer la DCT II sur la matrice entière, celle-ci est décomposée en blocs (généralement de taille 8 × 8 pixels) avant de transformer chaque bloc indépendamment.

Les formules des transformations directe et inverse de DCT II sont présentées ci-dessous :

Transformation directe:

$$F(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)cos\left[\frac{(2x+1)u\,\pi}{2N}\right]cos\left[\frac{(2y+1)v\,\pi}{2N}\right]$$

$$Pour\ \ u,v\ = 0,1,2,3,\dots,N-1$$

L'imagerie numérique/ JPEG / DCT

Transformée inverse :

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)\, F(u,v)\, cos\left[\frac{(2x+1)u\,\pi}{2N}\right] cos\left[\frac{(2y+1)v\,\pi}{2N}\right]$$

$$Pour\ \ x,y\ = 0,1,2,3,\dots,N-1, avec$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & si\ u = 0 \\ \sqrt{\frac{2}{N}} & si\ u \neq 0 \end{cases}$$

-f(x,y) représente une valeur de l'image initiale pour x et y données.

-F(u,v) représente les coefficients de la DCT II.

- N représente la taille d'un bloc.

L'imagerie numérique/ Compression EZW

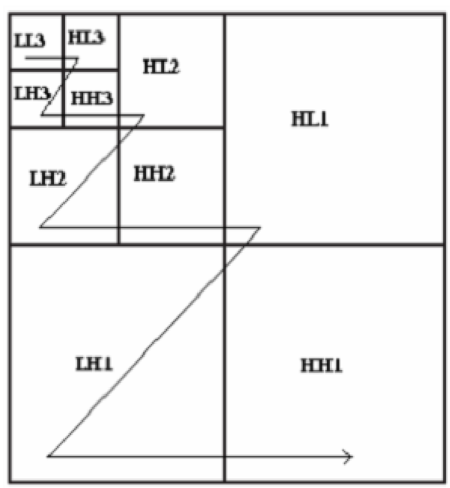
Le principe de l'algorithme EZW est de calculer la transformée en ondelette de l'image, puis on passe à l'étape de codage des coefficients transformés à l'aide d'une suite décroissante de seuils

$$T_0, \dots, T_{n-1} \text{ avec } T_{i+1} = \frac{T_i}{2}$$

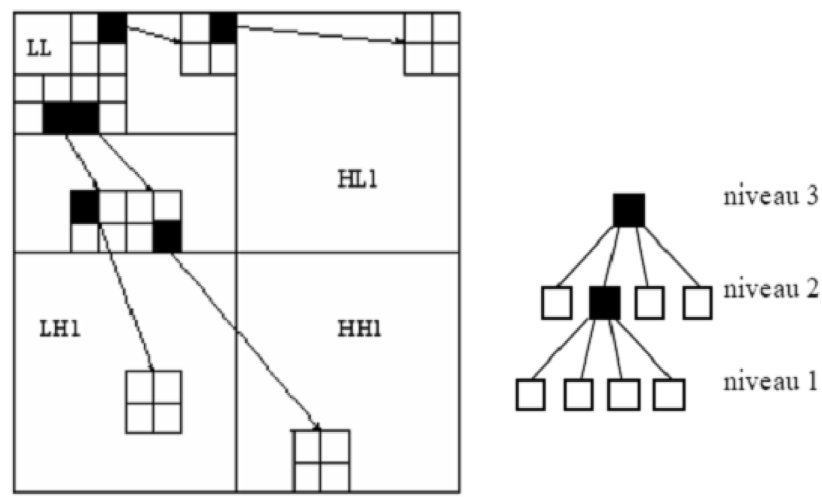
Pour coder les coefficients, l'algorithme effectue récursivement deux passes successives, ne traitant à chaque fois que les coefficients significatifs par rapport au seuil courant : ceux dont la valeur absolue est supérieure au seuil.

L'imagerie numérique/ Compression EZW

Dans la première passe, la dominante de l'algorithme parcourt les coefficients de la transformée en ondelettes suivant l'ordre donné par la figure suivante et pour la recherche des coefficients significatifs par rapport au seuil courant, en utilisant la hiérarchie donnée par la même figure.



ordre de parcours des coefficients



organisation hiérarchique des coefficients

L'imagerie numérique/ EZW / DTW

Les ondelettes c'est d'abord une théorie mathématique récente d'analyse du signal, développée dans les années 80.

L'analyse en ondelettes constitue une suite logique des travaux abordés dans le cadre de l'analyse temps-fréquence (échelle). L'analyse temps-échelle nous conduit à deux types de transformations en ondelettes : continues (TOC) et discrètes (TOD).

La transformée en ondelettes discrètes (TOD) est un outil mathématique qui décompose un signal x(u), en deux parties: approximation et détail. Le signal de départ est projeté sur un ensemble de fonctions de base qui s'adaptent aux fréquences du signal à analyser.

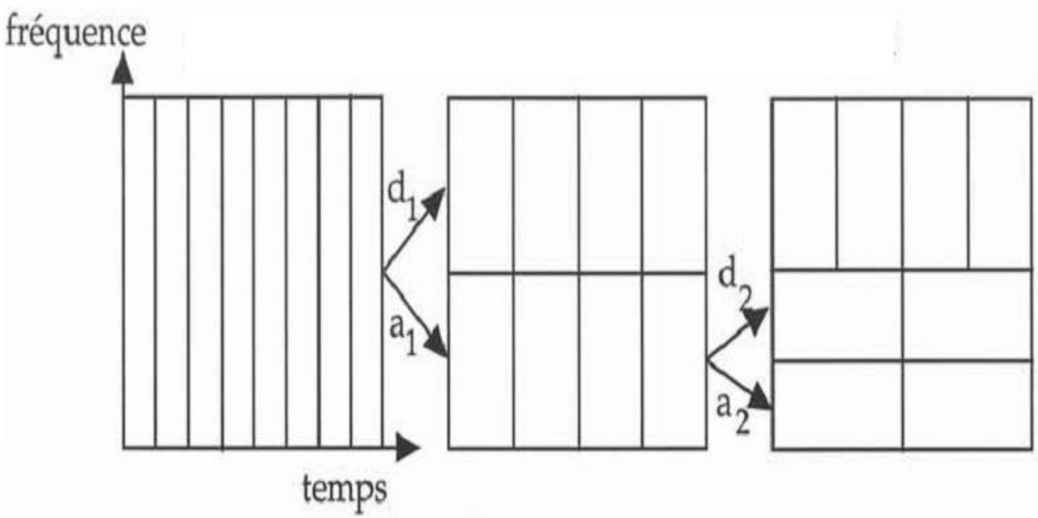
L'imagerie numérique/ EZW / DTW

L'approximation est obtenue en projetant le signal sur les translatées d'une fonction basse fréquence appelée fonction d'échelle. Cette projection réalise un filtrage passe-bas qui ne retient que les variations lentes du signal. Les détails sont obtenus en projetant le signal sur une fonction haute fréquence appelée ondelette. Cette projection réalise un filtrage passe-haut qui retient les évolutions rapides du signal.

Ces 2 opérations sont réversibles: le signal peut être reconstruit à partir de l'approximation et des détails.

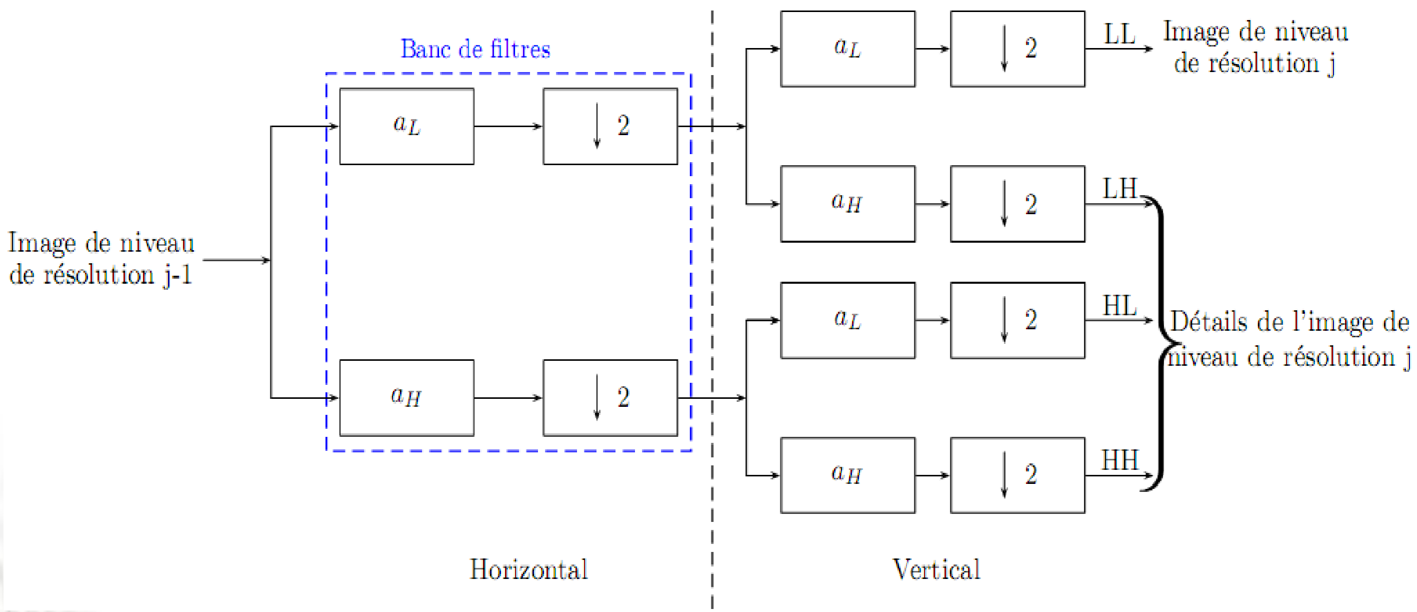
L'imagerie numérique/ EZW / DTW

Le signal d'approximation est ensuite décomposé en son approximation et ses détails, et ainsi de suite.



L'imagerie numérique/ EZW / DTW

Une transformée en ondelettes 2D possède trois fonctions d'ondelettes : la première fonction d'ondelettes permet de récupérer les détails horizontaux, la deuxième les détails verticaux et enfin la troisième les détails diagonaux. La DWT 2D peut être implantée en utilisant un banc de filtres, illustré sur la Figure suivante.



L'imagerie numérique/ EZW / DTW

Suite au passage de l'image à travers deux filtres, un filtre passe bas (H) et un filtre passe haut (L), on obtient la nouvelle image de la Figure suivante. Cette image est décomposable en quatre sous images distinctes HH, HL, LH et LL.

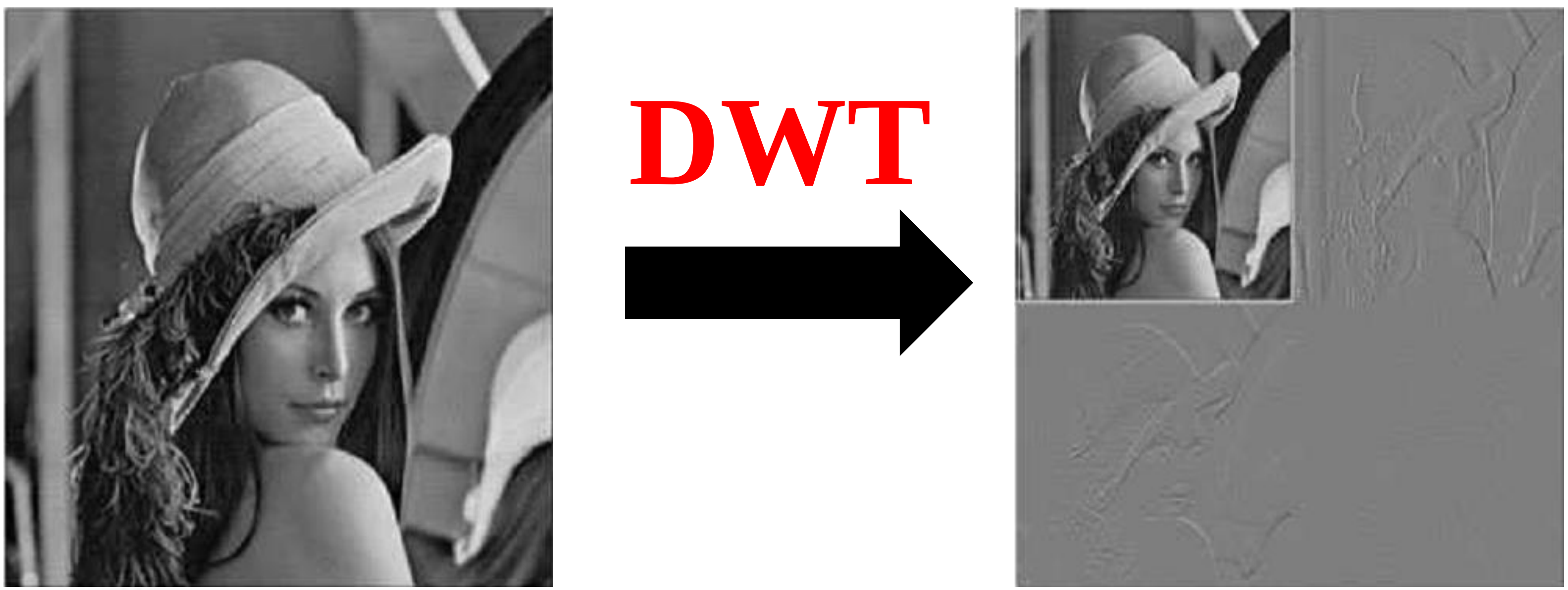
Chacune des sous images obtenues par filtrage contient une partie de l'information de l'image de départ. Le bloc LL contient une image de plus basse résolution (4 fois moins de pixels que l'image initiale).

LL1	HL1
LH1	HH1

Les blocs LH, HL et HH contiennent tous des détails de l'image, ou informations hautes fréquences.

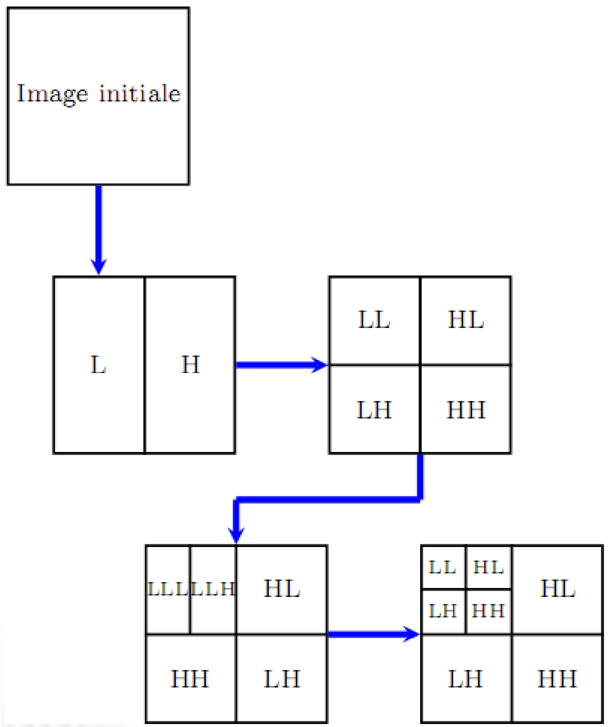
L'imagerie numérique/ EZW / DTW

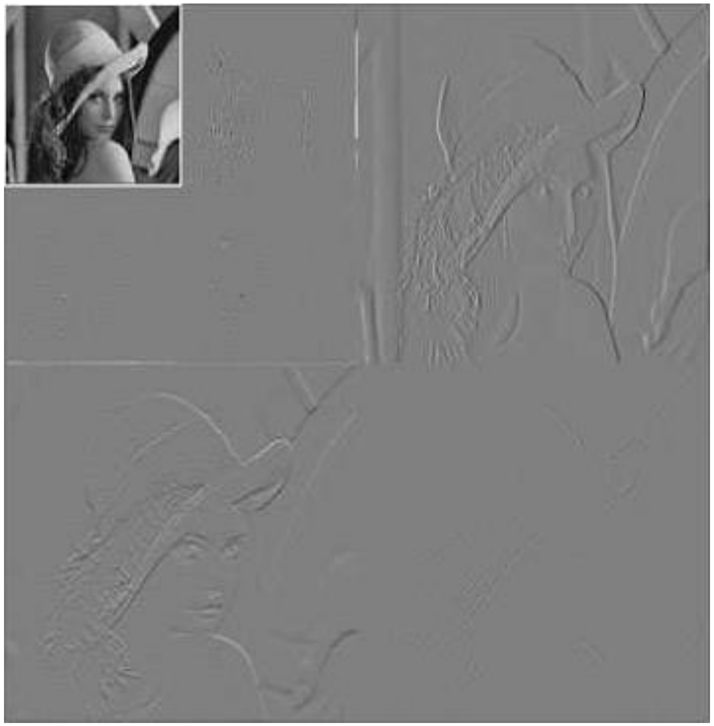
L'observation de l'image, permet d'illustrer les transformations subies par l'image Lena lors de l'application de la DWT 2D.



L'imagerie numérique/ EZW / DTW

Suite à ce premier niveau de décomposition, il est possible de poursuivre la décomposition en suivant l'algorithme de décomposition pyramidale de Mallat, comme montré dans Figure suivante





L'imagerie numérique/ EZW / DTW

Une ondelette est une fonction oscillante (ce qui explique le mot "onde"), appelée ψ de $L^2(R)$, qui a une valeur moyenne nulle $\int_{-\infty}^{+\infty} \psi(t)dt = 0$, et une durée limitée (ce qui explique le mot "ondelette", qui veut dire petite onde).

L'analyse en ondelettes adopte une fonction de prototype d'ondelettes connue sous le nom de "Ondelettes mère". L'ondelette mère ψ génère une famille d'ondelettes au temps b et à l'échelle a : $\left\{\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)\right\}_{(a,b)\in\mathbb{R}\times\mathbb{R}_0^+}$

L'imagerie numérique/ EZW / DTW

La transformée en ondelettes de $f \in L^2(R)$ au temps b et à l'échelle a :

$$wf_{a,b} = \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)dt$$

La transformée en ondelettes discrète (DWT) translate et dilate l'ondelette mère selon des valeurs discrètes. Ces coefficients a et b seront discrétisés de la manière suivante :

$a = a_0^j$ et $b = kb_0a_0^j$ avec $a_0 > 1 \; et \; b_0 > 0$ et j entier et k constante.

$$\psi_{a_0,b_0}(t) = \frac{1}{\sqrt{a_0^j}}\psi\left(\frac{t-kb_0a_0^j}{a_0^j}\right)$$

L'imagerie numérique/ EZW / DTW

Ainsi, la transformée discrète en ondelettes donnée par la formule ci-dessous :

$$DWT_{a_0,b_0}(t) = \frac{1}{\sqrt{a_0^j}} \int_{+\infty}^{-\infty} x(t)\psi\left(\frac{t - kb_0 a_0^j}{a_0^j}\right) dt$$

Il existe une infinité de fonctions d'ondelettes parce que toute fonction oscillante localisée est une ondelette mère possible. De nombreux spécialistes des ondelettes ont construit des familles d'ondelettes possédant certaines propriétés remarquables. Parmi les familles d'ondelettes seules deux types de transformations ont été choisis pour le format JPEG2000 :

- la transformation de Cohen-Daubechies-Fauvaue.
- la transformation de Le Gall.

L'imagerie numérique/ EZW / DTW

Les différentes familles d'ondelettes sont utilisées selon leurs propriétés en fonction du problème à résoudre.

La compression d'images est une application immédiate de la transformée en ondelettes bidimensionnelle. Les étapes pour compresser une image sont

1. **Transformations** par ondelettes.
2. **Quantification** : les valeurs des images de détails inférieures à un certain niveau sont éliminées, en fonction de l'efficacité recherchée. C'est cette étape qui introduit des pertes.
3. **Codage** des valeurs restantes. Les données restantes sont transmises à un encodeur entropie, c'est-à-dire à un algorithme de compression de données.

On a terminé avec l'imagerie numérique