

TABLE DES MATIERES

REMERCIEMENTS	i
TABLE DES MATIERES	ii
NOTATIONS ET ABREVIATIONS	vi
INTRODUCTION GENERALE	1
CHAPITRE 1 GÉNÉRALITÉS SUR L'IMAGE ET LE TRAITEMENT D'IMAGE	2
1.1 Introduction	2
1.2 Définition de l'image	2
1.3 Image numérique	2
1.4 Image vectorielle	3
1.5 Image matricielle	3
1.6 Image à niveau de gris	3
1.7 Couleur	4
1.7.1 Définition de la couleur	4
1.7.2 Longueur d'onde de la couleur	4
1.8 Image couleur	5
1.8.1 La lumière	5
1.8.2 L'espace de couleur RGB	5
1.8.3 L'espace de couleur XYZ	6
1.8.4 L'espace de couleur HSV	6
1.8.5 L'espace Lab	6
1.9 Colorimétrie et espace de représentation	7
1.9.1 Matrice de couleur	7
1.9.2 Perception psychovisuelle de l'image	8
1.10 Caractéristique de l'image	9

1.10.1	<i>Pixel</i>	9
1.10.2	<i>Dimension</i>	9
1.10.3	<i>Résolution</i>	10
1.10.4	<i>Bruit</i>	10
1.10.5	<i>Histogramme</i>	10
1.10.6	<i>Contour et texture</i>	11
1.10.7	<i>Luminance</i>	11
1.10.8	<i>Contraste</i>	11
1.10.9	<i>Région</i>	12
1.11	Système de traitement d'image	12
1.11.1	<i>Acquisition des données images</i>	12
1.11.2	<i>Prétraitement et post-traitement</i>	13
1.12	Domaine d'application	13
1.12.1	<i>Méthodes de traitement d'images</i>	14
1.12.2	<i>Opérateur statistique</i>	14
1.12.3	<i>Filtrage d'image</i>	15
1.13	Conclusion	17
CHAPITRE 2 PRINCIPE ET METHODES DE COMPRESSION D'IMAGE		19
2.1	Introduction	19
2.2	Compression d'une image fixe	19
2.2.1	<i>Principe</i>	19
2.2.2	<i>Synoptique général d'une chaîne de compression</i>	20
2.2.3	<i>Système de compression par transformée</i>	21
2.3	Quantification et codage	23
2.3.1	<i>Quantification</i>	23

2.3.2	<i>Codage</i>	28
2.4	Méthodes d'évaluation d'une chaîne de compression	34
2.4.1	<i>Outils de mesure d'évaluation</i>	34
2.4.2	<i>Différents types de mesure d'évaluation</i>	34
2.5	Conclusion.....	41
CHAPITRE 3 MODELE DE COMPRESSION D'IMAGE AVEC TKL		42
3.1	Introduction.....	42
3.1.1	<i>Historique</i>	42
3.1.2	<i>Transformation de Karhunen-Loeve</i>	43
3.1.3	<i>Catégories de KLT</i>	46
3.1.4	<i>Calcul des coefficients de TKL</i>	48
3.2	Modèle de compression par TKL	53
3.2.1	<i>Chaîne de compression avec TKL</i>	53
3.2.2	<i>Algorithme de la compression avec TKL</i>	54
3.3	La reconstruction de l'image issue de compression par TKL	57
3.4	Conclusion.....	58
CHAPITRE 4 SIMULATION ET EVALUATION DE LA METHODE DE COMPRESSION D'IMAGES AVEC TKLH		60
4.1	Introduction.....	60
4.2	Modèle de compression proposée avec la méthode TKL.....	60
4.3	Objectifs	61
4.4	Modèle de simulation	61
4.5	Résultats et évaluation de la simulation.....	62
4.6	Conclusion.....	67
CONCLUSION GENERALE		68

ANNEXE 1 NORME JPEG69
ANNEXE2 CODAGE D’IMAGE75
ANNEXE 3 LOGICIEL MATLAB81
BIBLIOGRAPHIE83
FICHE DE RENSEIGNEMENTS.....85
RESUME.....86
ABSTRACT86

NOTATIONS ET ABREVIATIONS

1. Minuscules grecs

θ Ecart type

λ Longueur d'onde

$\lambda_R, \lambda_G, \lambda_B$ Longueur d'onde du canal Rouge Vert Bleu

μ_x Moyenne de x

σ_x Ecart type

$\Psi(t)$ Fonction d'ondelette

2. Majuscules grecs

Λ Matrice de covariance

Γ_x Matrice de covariance pour la variable x

Δ Ensemble fini

3. Minuscules latines

d_0 Pas de quantification

d_1, d_2, d_L Intervalle de décision 1, 2, L

k Dimension du vecteur

m_x Moyenne d'une variable aléatoire

q Pas de quantification

p Pixel

p^*	Signal quantifié
$u(m)$	Signal analogique
v	Vitesse de la lumière
x	Valeur d'entrée des bruits ou erreurs
x_{min}	Niveau de gris minimal
x_{max}	Niveau de gris maximal
x_p	Amplitude du pixel de l'image originale

4. Majuscules latines

A_0	Image originale
C	Célérité de la lumière
C	Contraste
D	Gamme dynamique
D_{2H}	Dérivée second horizontale
D_{2V}	Dérivée second verticale
D_H	Dérivée horizontale
D_V	Dérivée verticale
$E(X_j)$	Esperance mathématique de X_j
IR, IG, IB	Matrice canal Rouge, Vert, Bleu de l'image
IR_{TKL}, IG_{TKL}	Matrice canal Rouge, Vert, Bleu de l'image par la transformé de K.L.

I_k	Matrice diagonale
$I(x, y)$	Coordonnées de l'image au point x, y
$I_{S1}(x, y)$	Image de sortie du système
K	Transformation de KL
L	Dynamique des valeurs des pixels
L	Valeur de reproduction
$L + 1$	Niveaux de décisions
L_1, L_2	Degrés de luminosité de deux zones voisines
M	Nombre de ligne
N	Négatif d'une image
N	Nombre de colonne
N_g	Niveau de gris
$P(x_i)$	Probabilité de x_i
Q	Fonction de quantification
$S(x, y)$	Structure des images
S_H	Dérivation horizontale
S_V	Dérivation verticale
S_O	Dérivation oblique
V	Matrice de la transformation inverse de Karhunen Loeve

$V(X, Y)$	Variance
w_i	Matrice de transformation en ondelette
X_{max}	Luminance maximale de l'image

5. Abréviations

ACP	Analyse en composantes Principales
BMP	Bit MaP
CCD	Charge Coupled Device
CIE	Compagnie Internationale de l'Eclairage
$Cor(X, Y)$	Corrélation entre les variables X, Y
$Cov(X, Y)$	Covariance entre les variables X, Y
dB	Décibel
EQM	Erreur Quadratique Moyenne
EQM	Erreur Quadratique Moyenne
HDTV	High Definition Television
JPEG	Joint Photographic Expert Group
Log	Logarithme népérien
LSB	Less Significant Bit
LZW	Lempel Ziv Welch
MATLAB	Matrix Laboratory

MIT	Massachussetts Institute of Technology
MSE	Mean Square Error
Pixel	Picture element
POD	Proper Orthogonal Decomposition
PSNR	Peak Signal to Noise Ratio
QS	Quantification scalaire
QV	Quantification Vectorielle
\mathbb{R}	Nombres réels
RGB	Red Green Blue
RLE	Run Length Encoding
RMSE	Root Mean Square Error
RVB	Rouge Vert Bleu
SFI	Système de Fonction Itérée
SFIL	Système de Fonction Itérée Local
SNR	Signal to Noise Ratio
SSIM	Structural SIMilarity
SVD	Singular Values Decomposition
SVH	Système Visuel Humain
TCD	Transformée en Cosinus Discrète
TIFF	Tagged Image File Format
TF	Transformée de Fourier
TKLH	Transformation Karhunen Loeve Huffman
TFD _U	Transformée fourrier unitaire
VA	Variable Aléatoire

INTRODUCTION GENERALE

Avec l'utilisation des appareils numériques tels que scanner, appareils photo, le besoin de stocker davantage se fait ressentir de plus en plus. Ainsi, les encyclopédies les plus complètes comportent environ 25000 mille pages, contenant des textes et des images. Si on estime à 1 Mo la taille d'une page scannée, l'espace de stockage total nécessaire sera alors de 25 Go.

Même s'il existe aujourd'hui des mémoires de stockage de masse de grande capacité, le besoin de stocker s'avère encore insatisfait. Une des solutions à ce problème est le recours à des techniques de compression d'images fixes.

Devant ces faits, des chercheurs ont concentré leurs études sur de nouvelles méthodes de compression d'images. Les plus classiques étaient les méthodes de compression basées sur les codages sans perte ou avec perte : par exemple la compression de Huffman, ou dans le domaine transformé tel que la DCT. Plus tard, il y avait JPEG, JPEG 2000.

Malgré les performances de la compression JPEG, elle n'est pas la meilleure car, entre autres, la transformation T.C.D. n'est pas optimale au sens de la décorrélation.

C'est pourquoi, on va essayer dans ce travail de remédier à ces inconvénients de JPEG en utilisant la TKL dans une chaîne de compression comme une transformation optimale. Ainsi, notre travail se porte sur « *La compression d'images fixe couleur avec la méthode la TKLH* ».

La transformée de Karhunen est une méthode optimale de codage des images au sens de l'erreur quadratique moyenne. On ne trouve dans la littérature qu'une méthode adaptative dans le domaine spatiale. De nombreuses autres méthodes de codage sous-optimales ont été développées pour éviter les problèmes liés à la transformée avec l'analyse factorielle de données. De telles méthodes ont des performances inférieures à la TKL bien du point de vue de la qualité visuelle que de l'erreur quadratique moyenne.

Le document est divisé en quatre parties, on verra en premier chapitre, les généralités sur les images et les traitements d'images ; puis, on va suivre dans le second chapitre avec les principes de base de la méthode de compression d'images.

Le troisième chapitre se porte sur les fondements mathématiques de la transformation de Karhunen Loeve ; enfin, on terminera dans le chapitre trois, avec la simulation de la compression d'image couleur utilisant la TKL.

CHAPITRE 1

GÉNÉRALITÉS SUR L'IMAGE ET LE TRAITEMENT D'IMAGE

1.1 Introduction

Le traitement d'image est une étape à la fois importante et secondaire. Importante, parce qu'il faut bien connaître les processus de traitement pour tirer la quintessence de l'image.

Secondaire parce que cette quintessence n'est pas créée par le traitement mais par la qualité de prise de vue. On ne dira jamais assez que la qualité d'une image est surtout le résultat d'étapes très classiques comme la collimation du télescope, la focalisation, le réglage de la webcam.

On ne sera donc pas étonné du peu de longueur de la partie consacrée à la révélation des détails, car si la prise de vue est réussie, il ne reste plus grand-chose à faire au cours du traitement.

D'une manière générale, plus une prise de vue est réussie, et moins l'image nécessite un traitement compliqué.

Les longs scripts de traitement sont toujours révélateurs d'une prise de vue non optimale (que ce soit à cause des erreurs de l'observateur ou bien des conditions d'observation qu'il ne maîtrise pas). Le traitement d'image ou encore amélioration des images pour l'interprétation par un observateur humain permet une meilleure visualisation de l'image

1.2 Définition de l'image

- Une image est un ensemble structuré d'informations qui, après affichage sur l'écran a une signification pour l'œil humain.
- Elle peut être décrite sous la forme d'une fonction $I(x, y)$ de brillance analogique, définie dans un domaine borné tels que x et y sont les coordonnées spatiales d'un point de l'image. Sachant que I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect l'image est inexploitable par ordinateur, ce qui nécessite sa numérisation [1].

1.3 Image numérique

- Une image numérique est obtenue après numérisation d'une image analogique (Echantillonnage, Quantification, codage).

- Une image numérique est une image dont la surface est divisée en éléments de taille fixe appelée *cellule* ou *pixel*. Chaque pixel ayant chacun comme caractéristique, *un niveau de gris* (N_g). Ce niveau de gris (pour l'image noir et blanc) ou couleur prélevée à l'emplacement correspondant dans l'image réelle ou calculée à partir d'une description interne de la scène à représenter.
- La numérisation d'une image est la conversion de celle-ci, c'est-à-dire de son état analogique en une image numérique représentée par une matrice bidirectionnelle de valeur numérique $I(x, y)$.

Mais dans ce cas-là, x et y : sont des coordonnées cartésiennes d'un point de l'image $I(x, y)$. et N_g : niveau de gris en ce point.

- Chaque élément $I(x, y)$ représente un pixel de l'image et sa valeur est associée à un niveau de gris codé sur n bits [1].

La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur.

1.4 Image vectorielle

- Une image vectorielle ou image en *mode trait*, est une image numérique composée d'objets géométriques individuels.
- Par exemple, un segment de droite, polygones, arcs de cercle. Ceci définit chacun par divers attributs de forme, de position de couleur.
- Elle se différencie en cela des images matricielles qui elles sont constituées de pixels. L'intérêt est de pouvoir redimensionner l'image à volonté sans aucun effet d'escalier [2].

1.5 Image matricielle

- Une image matricielle ou *carte de points*, en anglais « *bitmap* », est une image constituée d'une matrice de points colorés.
- Cette image matricielle est constituée d'un tableau, d'une grille, ou chaque case possède une couleur qui lui est propre et est considérée comme un point.

Il s'agit donc d'une juxtaposition de points couleurs formant, dans leur ensemble, une image. Les points de couleurs constituant l'image s'appellent des *pixels* [2].

1.6 Image à niveau de gris

- Le niveau de gris est la valeur de l'intensité lumineuse en un point.

- La couleur du pixel peut prendre des valeurs allant du noir ou blanc en passant par un nombre fini de niveau intermédiaire [0 255].
- Pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée.
- Chaque pixel est codé sur 8 bits et peut prendre une valeur comprise entre 0 et 255. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.
- Le nombre de N_g dépend du nombre de bits utilisé pour décrire la couleur de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux accessibles sont nombreux [1].

1.7 Couleur

1.7.1 Définition de la couleur

- On appelle couleur la perception faite par la vision de l'intensité de certaines longueurs d'ondes de la lumière visible.
- La sensation de couleur prend son origine dans la stimulation de cellules nerveuses spécialisées nommées cônes situés au fond de l'œil [3].

1.7.2 Longueur d'onde de la couleur

- La couleur d'une radiation monochromatique dépend de sa fréquence, donc la couleur dépend aussi de sa longueur d'onde puisque $\lambda = \frac{c}{\nu}$.
- De ce fait, on peut dire que la fréquence ne change pas lorsque la lumière passe d'un milieu transparent à un autre. Mais par contre sa longueur d'onde, elle change.

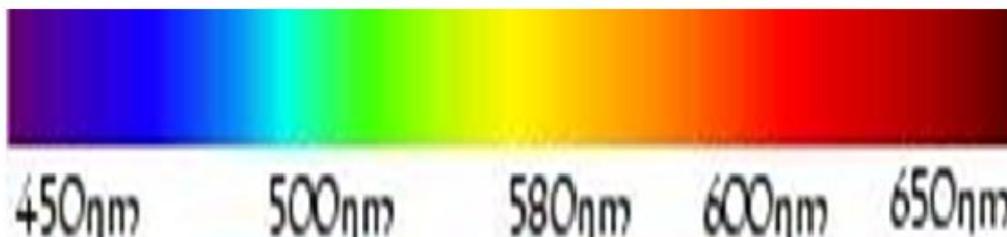


Figure 1.01 : Correspondance de longueurs d'ondes

1.8 Image couleur

- On appelle couleur, la perception subjective que fournit l'œil d'une ou plusieurs fréquences d'ondes lumineuses, avec une ou des amplitudes données.
- C'est aussi la composition de l'intensité de chacun des trois canaux des couleurs fondamentales : le rouge R, le vert V, et le bleu B [4].

$$f : i^2 \rightarrow [0,255]^3 \quad (1.01)$$

$$(x, y) \rightarrow f(x, y) = (R(x, y), V(x, y), B(x, y)) \quad (1.02)$$

1.8.1 La lumière

On appelle la lumière, une énergie électromagnétique comportant une partie ondulatoire (onde) et une partie corpusculaire (les photons).

1.8.2 L'espace de couleur RGB

L'espace RGB (Red, Green, Blue) défini par la compagnie internationale de l'éclairage (C.I.E.) en 1931 représente les couleurs par synthèse soustractive.

Chaque couleur est représentée par trois composantes :

- le rouge (R),
- le vert (V) et
- le bleu (B)

Dont les longueurs d'onde normalisées sont respectivement : $\lambda_R, \lambda_G, \lambda_B$.

$$-\lambda_R = 700 \text{ nm}$$

$$-\lambda_G = 546.1 \text{ nm}$$

$$-\lambda_B = 435.8 \text{ nm}$$

En général, les trois composantes sont représentées sur l'ordinateur par trois octets (24 bits). Avec ce choix, pour le stockage des couleurs, on obtient une palette de 256^3 soit 16 777 216 couleurs possibles.

Le principal inconvénient de cet espace de représentation vient du fait qu'il ne tient pas compte de la variation de la sensibilité de l'œil.

1.8.3 L'espace de couleur XYZ

- La C.I.E. a défini un espace de représentation qui prend en compte la sensibilité de l'œil. Il s'agit de l'espace XYZ.
- L'espace XYZ est rarement utilisé dans les recherches d'images car il n'est pas perpétuellement uniforme. C'est-à-dire que les différentes nuances d'une même teinte, par exemple rouge peuvent être très éloignées dans l'espace.

1.8.4 L'espace de couleur HSV

- L'espace HSV (Hue Saturation Value), en français, teinte, saturation et valeur (luminance) possède une composante de luminance, une composante de teinte qui correspond à la couleur et une saturation de cette teinte (correspondant à la pureté de la couleur).
- Cet espace présente l'avantage de simuler le comportement visuel humain dans lequel la couleur est décomposée en une image de luminance (la valeur) et une image de chrominance (teinte et saturation).
- La teinte fournit directement l'information de la couleur dominante dans les différentes régions de l'image.
- L'espace HSV n'est malheureusement pas uniforme, c'est-à-dire qu'une distance entre couleurs visuellement proches calculée dans cet espace peut être très grande.

1.8.5 L'espace Lab

L'espace Lab est un espace couleur perpétuellement uniforme, c'est à dire qu'il possède la bonne propriété de respecter les distances entre couleurs visuellement proches.

Il est défini à partir de l'espace XYZ par des relations non linéaires.

1.9 Colorimétrie et espace de représentation

- Un espace couleur, ou système de représentation, se définit par le choix des primaires utilisées et du blanc de référence qui fixe leurs valeurs unitaires.
- Il peut être défini quatre familles de systèmes de représentation :
 - o les systèmes de primaires,
 - o les systèmes luminance-chrominance,
 - o les systèmes liés à la perception humaine et
 - o les systèmes d'axes indépendants
- Des versions normalisées de ces systèmes sont proposées par la CIE. Les données étant définies dans un espace de dimension supérieure à 1, il est assez naturel de se poser la question du choix de l'espace dans lequel les traitements vont être effectués.

Faut-il travailler dans l'espace initial où sont obtenues les données, ou, au contraire, faut-il chercher un espace mieux adapté aux traitements envisagés.

La réponse à cette question n'est pas unique et dépend bien sûr de la nature des images et des traitements que l'on désire effectuer.

Les deux raisons essentielles suscitant la recherche d'un nouvel espace de travail sont :

- la volonté de réduire la dimensionnalité de l'espace initial des données ou bien
- la recherche d'un espace mettant en relief les informations nécessaires à l'élaboration des traitements.

La réversibilité de la transformation permettant de se projeter dans le nouvel espace, peut être un élément guidant le choix de la transformation, car elle permet, après traitement, de retrouver une image de même nature que l'image initiale [5].

1.9.1 Matrice de couleur

On peut toujours représenter l'image sous forme matricielle que nous noterons I .

Cette matrice est de dimension trois notée I_{m*n*3} où m est la largeur, n est la longueur et 3 est le nombre de « couleurs de base » utilisées.

Ainsi, une image est représentée par trois matrices de dimension $m * n$.

Le nombre total de bits nécessaire pour représenter l'image est encore en fonction du nombre de bits représentant chaque pixel. Et ce dernier, est fonction du nombre de bits utilisés pour caractériser chaque couleur de base noté N_c . Comme formule générale, nous avons alors :

$$\text{Nombre total de bits} = m * n * 3 * N_c \quad (1.03)$$

Avec :

N_c : Nombre de bits caractérisant chaque couleur de base

1.9.2 Perception psychovisuelle de l'image

- Cette notion est vue ici dans le cadre de la compression qui vise toujours à améliorer le rapport entre le taux de compression et les dégradations de l'image (compromis taux/distorsion).
- Le problème est de savoir jusqu'à quelles limites, les dégradations sont permises pour qu'elles ne soient perceptibles par l'œil humain, modélisé avec le système SVH (Système Visuel Humain).
- Dans certains cas, comme par exemple, dans des domaines spécifiques tels que l'imagerie médicale, il se peut que des détails très fins (exemple : fissure d'un os) qui ne sont même pas perceptibles par l'œil humain mais qui doivent être conservés pour ne pas fausser les diagnostics.
- Cette formulation du problème nécessite encore des outils appelés « *métriques perceptuelles* » adéquats qui permettent de mesurer les dégradations et de juger ces valeurs si elles sont admissibles ou non par rapport au système SVH et aux enjeux de la compression.
- La recherche de solution revient toujours à exploiter les propriétés psychovisuelles de l'œil humain qui est sensible :
 - aux variations de l'intensité lumineuse,
 - à la répartition spatiale des fréquences ou
 - au changement brusque du contenu.

Ces propriétés sont les suivantes :

- L'œil est beaucoup plus sensible à la variation de l'intensité de la lumière (luminance) qu'à la variation des couleurs (chrominance).

Cette propriété a été exploitée en compression pour éliminer beaucoup plus d'informations dans les canaux de couleur en faisant un sous échantillonnage.

- Des expérimentations ont démontré que l'œil s'avère beaucoup moins sensible à des variations de luminance et de chrominance dans des zones de type frontières (hautes fréquences) que dans des zones uniformes ou homogènes (basses fréquences).

Cette propriété peut être exploitée en compression pour éliminer beaucoup plus d'informations dans des zones homogènes que dans des zones contenant des informations pertinentes (textures, contours).

- Le système de vision humain a tendance à accentuer les contrastes pour permettre une meilleure visibilité des couleurs. Ainsi, un carré gris neutre sur un fond blanc n'apparaîtra pas de la même intensité lumineuse qu'un carré gris neutre sur un fond noir.

1.10 Caractéristique de l'image

1.10.1 Pixel

Comme il a été dit dans les paragraphes ci-dessus, une image est constituée d'un ensemble de points appelés *pixels*.

Le mot *pixel* provient d'une contraction de l'expression anglaise « *PICTure ELement* ».

Le pixel représente le plus petit élément constitutif d'une image numérique auquel on peut associer individuellement une couleur (ou un N_g) et une intensité.

Chaque pixel est défini dans une image par des coordonnées et étant donné que l'écran effectue un balayage de gauche à droite et de haut en bas, on désigne généralement par les coordonnées (0,0) le pixel situé en haut à gauche de l'image [1].

1.10.2 Dimension

On appelle *dimension* d'une image, la taille de l'image ou le nombre de pixel d'une image $[M * N]$ avec M le nombre de ligne de l'image, et N son nombre de colonne.

1.10.3 Résolution

On appelle *résolution* de l'image, la clarté ou la finesse de détail atteinte par un moniteur ou une imprimante dans la production d'image.

Sur les moniteurs d'ordinateur, la résolution est exprimée en nombre de pixel par unité de mesure.

On utilise aussi, le mot résolution pour désigner le nombre total de pixel affichable horizontalement sur un moniteur, plus grand est ce nombre meilleure est la résolution [1].

1.10.4 Bruit

On appelle « bruit », tout phénomène imprévisible qui vient perturber le signal.

Dans une image, c'est un phénomène de brusques variations d'intensité d'un pixel par rapport à ses voisins. Le bruit peut être causé par :

- Les évènements inattendus lors de l'acquisition comme le bougé ou une modification ponctuelle des conditions d'éclairage ;
- La mauvaise qualité des capteurs ou une mauvaise utilisation de ces derniers ;
- Lors de l'échantillonnage, le passage de la forme analogique à la forme numérique de l'image ;
- Ou bien la nature de la scène elle-même (Poussière, rayures, perturbations, atmosphérique) [6].

1.10.5 Histogramme

- L'histogramme des N_g ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris ou couleur dans une image.
- Pour diminuer l'erreur de quantification ou pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'*histogramme* correspondant.
- Un histogramme peut être utilisé pour améliorer la qualité d'une image (rehaussement d'images) en introduisant quelque modification pour pouvoir extraire les informations utiles de celle-ci.

1.10.6 Contour et texture

- Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative.
- Les textures décrivent la structure de ceux-ci.
- L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

1.10.7 Luminance

- On appelle la luminance, le degré de luminosité des points de l'image.
- Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface.
- Pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.
- La moyenne ou luminance (brillance) d'une image numérique en niveau de gris est définie comme la moyenne des pixels de l'image :

$$Lum(I) = \frac{1}{M * N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) \quad (1.04)$$

Avec :

M : Nombre de ligne

N : Nombre de colonne

$I(x, y)$: La valeur de niveau de gris au point

1.10.8 Contraste

On appelle contraste, l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image.

Le contraste est défini en fonction des luminances de deux zones d'images.

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2} \quad (1.05)$$

1.10.9 Région

La région est un ensemble de pixels connexes et homogènes.

Un pixel appartient à une région donnée s'il vérifie les caractéristiques de celle-ci (intensité lumineuse, niveau de gris,...). Une région est limitée par un contour.

1.11 Système de traitement d'image

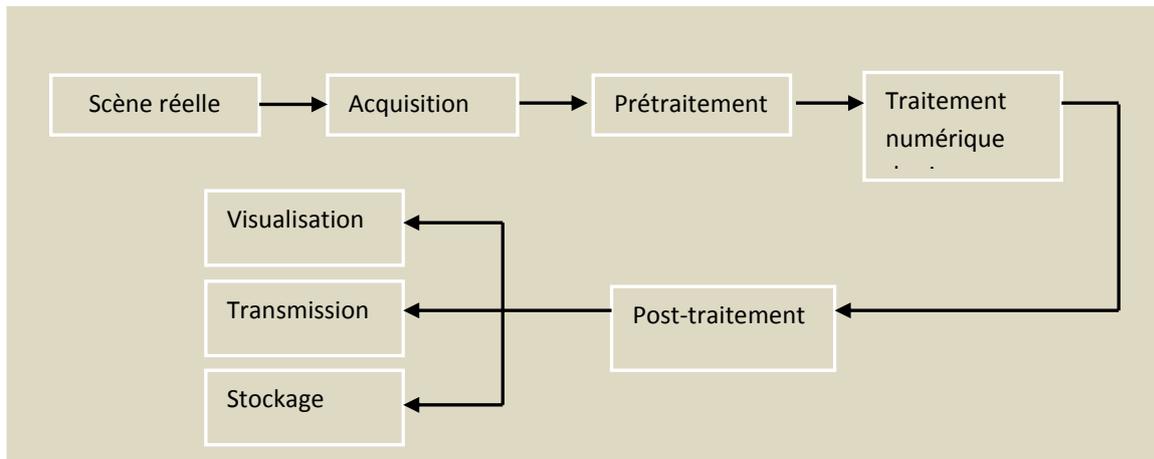


Figure 1. 02 : Système de traitement d'image

Avant d'être visualisé, une image passera dans quatre grandes étapes comme montré sur la figure 1.02.

1.11.1 Acquisition des données images

- L'acquisition des images constitue un des maillons essentiels de toute chaîne de conception et de production d'images.

Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système.

Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à *une procédure de numérisation*.

- Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales :

- les caméras numériques et
- les scanners.

A ce niveau, notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement d'image comme la télédétection, les arts graphiques, la médecine.

- Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur CCD (Charge Coupled Device).
- La carte d'acquisition reçoit les images de la camera, de la TV ou du scanner afin de les convertir en informations binaires qui seront stockées dans un fichier.

1.11.2 Prétraitement et post-traitement

Le *prétraitement* concerne les images acquises tandis que le *post traitement* concerne les images traitées.

Dans les deux cas, les images ne peuvent échapper aux effets de dégradations dus essentiellement aux phénomènes physiques tels que :

- La diffraction (déviation) du système optique.
- Le flou dû au mouvement de l'image durant son acquisition.

Pour pallier à ces dégradations on utilise en général le filtrage.

1.12 Domaine d'application

Le traitement d'images possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que :

- la médecine (radiographies,...),
- la biologie, météorologie,
- l'astronomie,
- la géologie,
- la physique (spectroscopie, physique des plasmas,...),
- les applications militaires,

- les applications industrielles (Robotique, surveillance de qualité,...),
- la photographie,
- la publicité.

1.12.1 Méthodes de traitement d'images

On appelle « traitement d'image », l'ensemble des méthodes et techniques opérant sur l'image.

Le traitement d'image a pour but d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.

1.12.2 Opérateur statistique

La description d'une forme nécessite un certain nombre d'opération qui transforme l'image originale en une représentation compatible au traitement ultérieur.

Cette représentation donne les caractéristiques d'une image exprimée sous forme numérique ou structurelle.

1.12.2.1 Transformations des niveaux de gris d'une image

On considère les images à N_g niveau de gris compris entre 0 et M (avec $M = 255$ par exemple).

Pour traiter le contraste d'une image, on applique uniformément à tous les pixels de celle-ci une transformation de niveau de gris f .

En d'autres termes, à partir d'une image I associant à tout pixel p le niveau de gris $I(p)$.

On obtient une nouvelle image J dont le niveau de gris au pixel p est $J(p) = f[I(p)]$.

1.12.2.2 Étirement de contraste

Supposons que, dans une image donnée, l'intervalle des niveaux de gris utilisé ne s'étend pas de 0 à M .

Soit g_{min} et g_{max} , les niveaux de gris minimum et maximum des pixels de l'image.

L'étirement de contraste consiste en application, aux niveaux de gris de l'image, d'une fonction linéaire f tel que : $f(g_{min}) = 0$ et $f(g_{max}) = M$

On a ainsi dans la formule (1.06).

$$f(x) = \frac{M(x - g_{min})}{g_{max} - g_{min}} \quad (1.06)$$

1.12.3 Filtrage d'image

- Une amélioration de l'image est essentiellement obtenue par une opération de filtrage.
- Le filtrage est utilisé pour modifier ou pour mettre en valeur une image.

Par exemple : on pourra filtrer une image pour accentuer certains attributs ou pour en supprimer d'autres.

- L'objectif du filtrage est d'éliminer les perturbations induites par les procédés d'acquisition d'image et au problème de transmission ainsi de réduire les variations d'intensité au sein de chaque région de l'image.

1.12.3.1 Filtrage linéaire

Le filtrage linéaire d'une image peut s'envisager de deux manières : le filtrage peut tout d'abord se réaliser dans :

- le domaine spatial en effectuant un produit de convolution, ou
- le domaine fréquentiel, en multipliant la transformée de Fourier de l'image par la fonction de transfert du filtre. L'image originale est alors obtenue par la transformée inverse.

1.12.3.2 Filtrage par produit de convolution

Comme dans le cas d'un système unidimensionnel, on peut interpréter l'action d'un opérateur bidimensionnel linéaire et invariant en translation comme un filtrage.

Si $I_{s1}(x, y)$ et $I_{s2}(x, y)$ sont les images de sortie du système relative aux images d'entrée $I_{e1}(x, y)$ et $I_{e2}(x, y)$; un système linéaire invariant dans le temps vérifie les propriétés suivantes :

- La linéarité

- L'invariance

Linéarité :

$$\alpha I_{e1}(x, y) + b I_{e2}(x, y) \rightarrow \alpha I_{s1}(x, y) + b I_{s2}(x, y) \quad (1.07)$$

Invariance :

$$I_e(x - t, y - t) \rightarrow I_s(x - t, y - t) \quad (1.08)$$

- Un filtre linéaire est un exemple important du système linéaire. Il est utilisé pour modifier le contenu spectrale d'une image afin d'en retirer l'information recherchée.
- Un filtre linéaire bidimensionnel est caractérisé par sa réponse impulsionnelle $h(x, y)$. L'image de sortie $I_s(x, y)$ résulte du produit de convolution de la réponse impulsionnelle du filtre par l'image d'entrée $I_e(x, y)$.

$$I_e(x, y) \rightarrow I_s(x, y) = I_e(x, y) * h(x, y) \quad (1.09)$$

- Contrairement à la manipulation d'histogramme, qui est une opération ponctuelle sur tous les pixels de l'image ; le filtrage est un produit de convolution qui met en jeu l'environnement de chaque pixel voisin.
- La convolution est définie par les relations suivantes :

A une dimension :

$$I_s(t) = I_e(t) * h(t) = \int_{-\infty}^{+\infty} I_e(\tau) h(t - \tau) d\tau \quad (1.10)$$

A deux dimensions :

$$I_s(x, y) = I_e(x, y) * h(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} I_e(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta \quad (1.11)$$

En ce qui concerne la convolution dans le domaine de traitement d'image, chaque pixel est remplacé par une combinaison linéaire des pixels qui l'entoure.

Un produit de convolution est un opérateur mathématique qu'on utilise pour multiplier des matrices entre elles. Dans le cas qui nous intéresse, nous mettons en jeu deux matrices très différentes :

- Une grande matrice
- Une petite matrice

La matrice image très grande et une matrice plus petite qu'on appelle le noyau parce que c'est le cœur de tout le changement qui affectera l'image et ce noyau est représenté par la relation suivante :

$$\begin{aligned}
 I_s(x, y) &= \sum_u \sum_v h(u, v) I_e(x - u, y - v) \\
 &= \sum_u \sum_v I_e(u, v) h(x - u, y - v)
 \end{aligned}
 \tag{1.12}$$

Avec :

- $h(x, y)$: Noyau du filtre
- $I_e(x, y)$: Image à filtrer

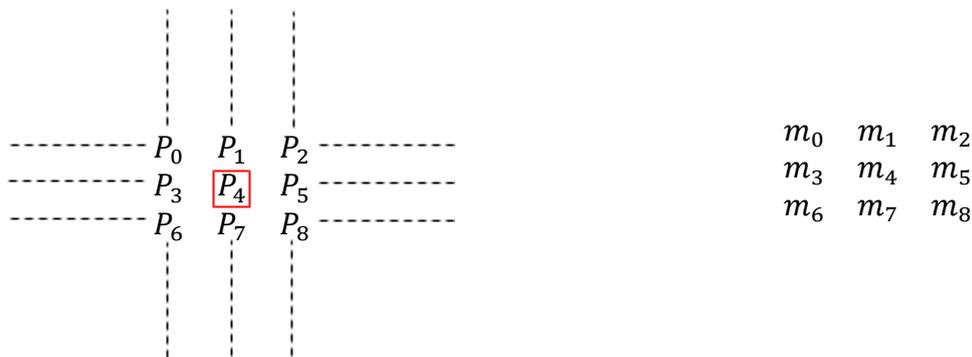


Figure 1.03 : Principe du filtrage d'image

En appliquant la convolution : P_4 devient $P_4 = m_0P_0 + m_1P_1 + \dots + m_8P_8$

Pour la convolution des bords, il y a l'ajout de zéro, l'enroulement, réflexion par défaut.

1.13 Conclusion

Avant de manipuler une image afin de la traiter, il est nécessaire de connaître ces caractéristiques, son concept et tous les aspects lui concernant.

On a vu alors les définitions concernant l'image, comme ses propriétés physiques internes à savoir les intensités lumineuses, les niveaux de gris ; ainsi que les traitements qu'elle peut subir.

Etant donné que l'image qu'on manipule est une image couleur, alors on a aussi parlé de la colorimétrie et des ondes de couleur.

Comme notre travail est basé sur la compression d'image, qui est une discipline dans le traitement d'image, on a entamé préalablement aussi sur les méthodes générales des traitements d'images. On verra dans le second chapitre, les synoptiques de base de la méthode de compression et les étapes à suivre pour sa réalisation.

Au cours du chapitre qui suit, nous présenterons la notion globale de codage et ses différentes variantes.

CHAPITRE 2

PRINCIPE ET METHODES DE COMPRESSION D'IMAGE

2.1 Introduction

Le but de la compression, c'est de compresser des données (son, images, vidéo). Le but est de réduire la quantité d'éléments binaires représentant l'information "image", c'est à dire diminuer la taille des données, et donc généralement la taille des fichiers.

Ceci, dans le but d'archiver ou transmettre les données d'une façon pertinente. En traitement d'images, il s'agit de minimiser autant que possible la quantité d'information (nombre de bits) nécessaire à une représentation fidèle de l'image originale.

Ceci est d'autant plus réalisable que la corrélation des pixels de l'image est importante.

Dans le cadre de la compression d'image couleur fixe, beaucoup de recherches sont encore entreprises pour satisfaire les besoins en matière de stockage et de transmission d'image.

2.2 Compression d'une image fixe

2.2.1 Principe

Dans la compression, l'essentiel est de concentrer l'énergie sur peu de coefficients, de façon à créer de grandes plages de zéros après quantification.

La compression réduit le nombre de données à transmettre. Pour certains signaux, elle permet, en outre, d'éliminer les redondances existantes afin de réduire la quantité d'information nécessaire à leurs représentations.

Plusieurs types de redondances en terme de corrélation peuvent être considérés :

- *Les redondances spatiales*, existantes dans toute image naturelle, entre pixels ou blocs voisins.
- *Les redondances spectrales*, apparaissant lorsque l'image est décomposée en plusieurs bandes spectrales ou sous-bandes. Les pixels situés alors sur une même position dans deux sous-bandes spectrales différentes sont souvent corrélés.
- *Les redondances temporelles*, entre images successives dans une séquence vidéo.

- *Les redondances structurales*, entre des blocs de pixels à diverses résolutions. Ces redondances sont exploitées avec profit dans un schéma de compression des images selon une approche fractale. Cette méthode opère sur une partition de l'image.

L'élimination des redondances spatiales et spectrales est souvent réalisée par passage dans un espace transformé.

2.2.2 *Synoptique général d'une chaîne de compression*

2.2.2.1 Système de compression

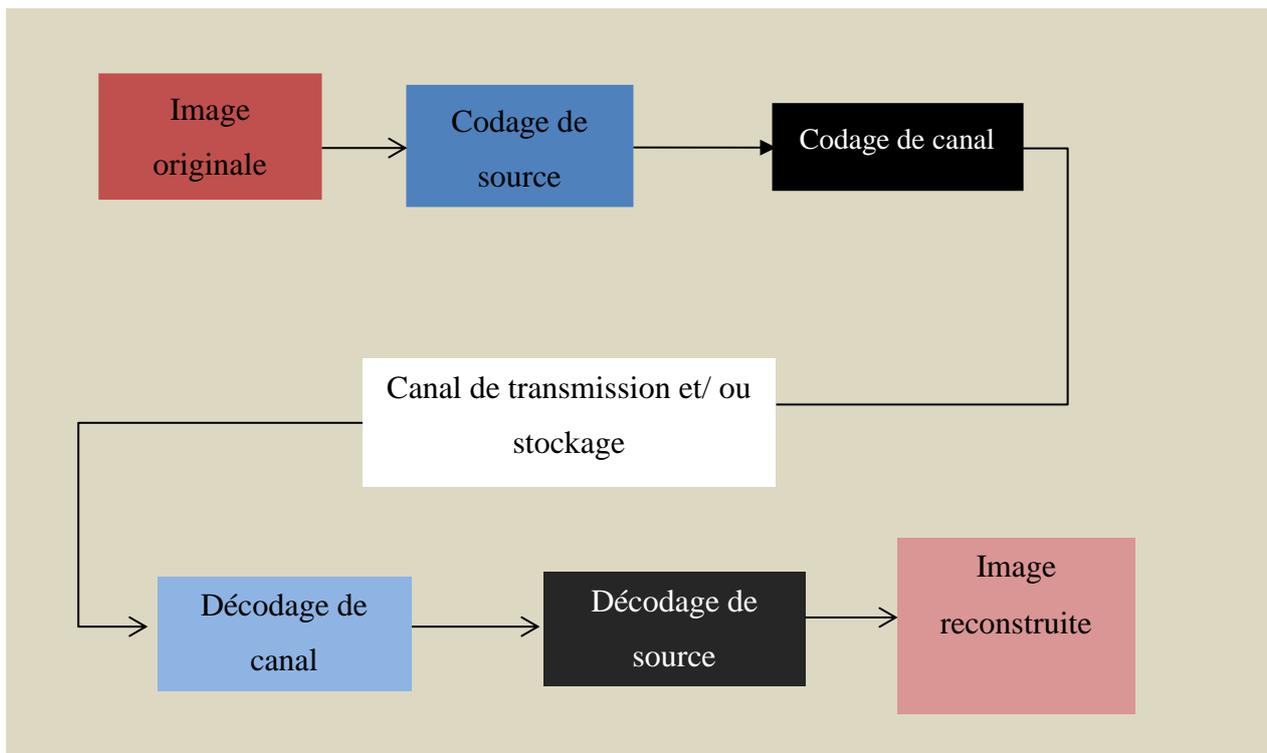


Figure 2.01: *Schéma synoptique d'une chaîne de compression*

2.2.2.2 Classification de codage

On distingue deux classifications de codage dans le système représenté dans la figure 2.01: le codage canal et le codage source [1], [7], [8], [9] [30].

- Le codage canal:

C'est l'adaptation du signal par rapport au canal pour avoir un taux d'erreur faible.

- Le codage source, on note :
 - a) Le codage sans perte (entropique) avec un taux de compression faible (1,5 à 2).
 - b) Le codage avec perte (irréversible), c'est à dire une suppression des redondances ou des informations inutiles car ce type de codage est prévisible et cette suppression est invisible par le système visuel humain (SVH).

Pour reconstruire l'image, on passe à l'étape de décodage inverse où on trouve les décodages de source et de canal.

2.2.3 Système de compression par transformée

On trouve la compression par transformée dans le domaine spatial; une chaîne de compression nécessite parfois une transformation pour rendre efficace les opérations de quantification et de codage. La figure 2.02 suivante donne la synoptique générale d'une chaîne de compression avec une transformation :

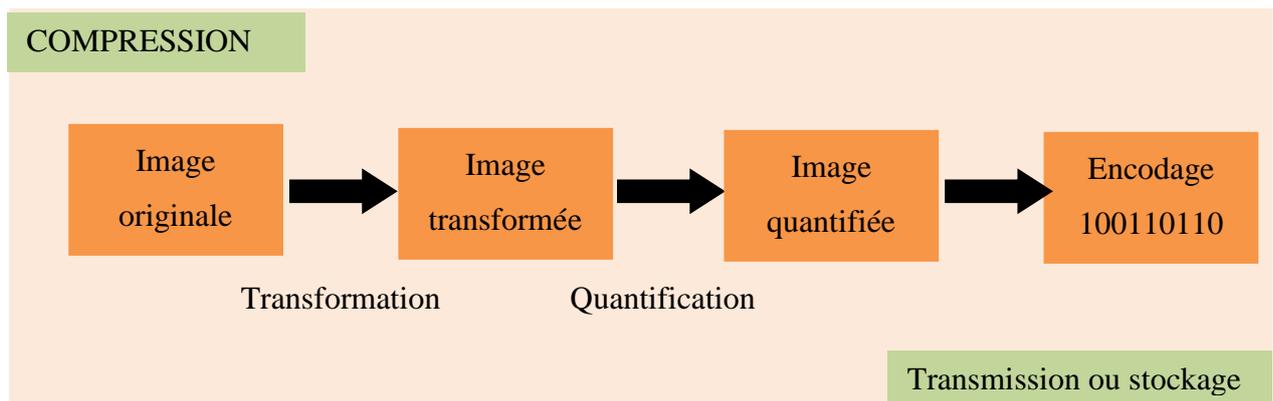


Figure 2.02: Chaîne de compression d'image par transformée

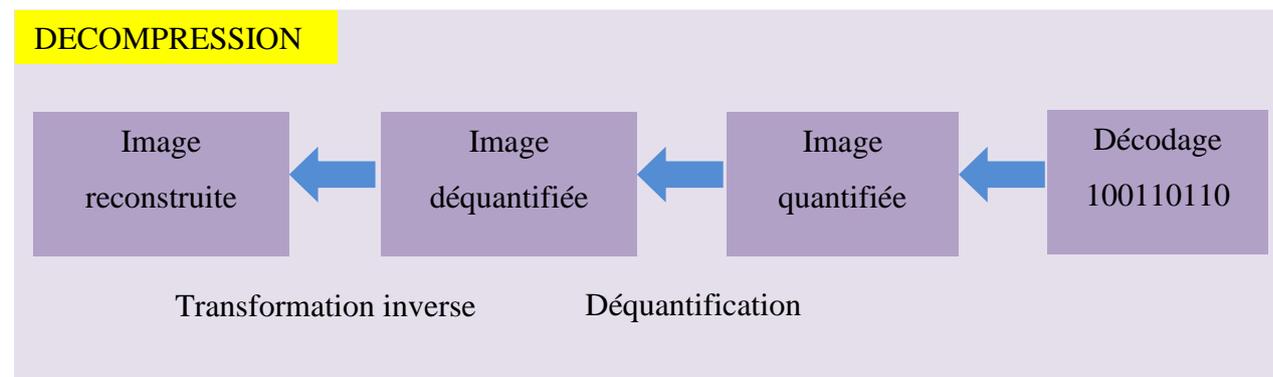


Figure 2.03: Chaîne de décompression d'image par transformée

La compression peut être considérée comme un problème d'optimisation, et pour ce faire, deux démarches peuvent être utilisés [7], [10].

- L'étape qui se trouve en amont de la quantification, consiste à la réalisation de la transformation de l'image. On distingue deux types de transformations:
 - o La transformation visant à supprimer les corrélations en tenant compte du signal traité. Cette méthode fait l'objet d'une procédure d'optimisation par arbre de décision et permet d'éliminer les redondances spectrales afin de réduire les redondances spatiales.
 - o En outre, on peut trouver aussi la transformation cherchant au contraire, à accroître les corrélations de façon à rendre efficace les étapes de quantifications et le codage.
- La seconde démarche consiste à figer la transformée et à développer une méthode de quantification et/ou de codage capables d'exploiter les corrélations résiduelles de la transformée. Ici donc, l'objectif des algorithmes de compression (quantification et codage) est l'exploitation de ces corrélations spatiales et spectrales dans l'image. On fige la transformée de façon à concentrer l'énergie sur peu de coefficients.

Notons en fait que, quelle que soit la démarche adoptée, le principe reste le même, c'est à dire que les opérations d'optimisation n'affectent pas qu'un seul bloc, mais résultent d'une bonne intégration de blocs dans la chaîne et d'une adéquation de méthodes et d'algorithmes aux caractéristiques globales du système de compression (nature des images, objectifs, contraintes...).

En effet, il n'existe ni méthode d'optimisation universelle ni méthode de compression parfaite mais au contraire, deux méthodes peuvent être basées sur des opérations inverses l'une de l'autre.

Exemple :

Une méthode supprimant les corrélations tandis que une autre tendant à les accroître, sachant toutes les deux visent le même objectif.

Dans notre travail, la méthode adoptée s'inscrit au niveau de la première démarche. C'est à dire réduire les redondances spatiales en supprimant les corrélations (corrélations nulles) et à choisir

le nombre minimum de coefficients décorrélés qui puissent donner une approximation optimale de l'image originale.

En effet, une compression dans le domaine de transformé présente certains avantages qui justifient notre choix:

- Si dans un espace transformé, les corrélations sont nulles (décorrélées), alors les opérations de quantification et de codage entropique sont bien simplifiées.
- Pour le cas d'une image, les valeurs des pixels dépendent les unes des autres, dans le sens où la valeur d'un pixel est fortement liée à celles de ses voisins, d'où l'importance de la décorrélation induite par la transformée qui rend l'optimisation de la quantification et du codage entropique des variables indépendantes plus simple que dans le cas sans décorrélation.
- Parmi les critères de choix d'une meilleure transformée est celui de la concentration d'énergie qui a pour effet de représenter les informations dans une image par un minimum de coefficients.
- Avec ce nombre réduit de coefficients associés à un codage entropique performant (ou peut être optimisé), on pourra réaliser une compression efficace.

2.3 Quantification et codage

2.3.1 Quantification

La représentation numérique d'un signal implique la quantification de chaque échantillon selon un nombre fini de valeurs discrètes.

L'objectif visé est, soit une transmission, soit un traitement déterminé (filtrage, analyse spectrale):

- Dans le premier cas, chaque échantillon du signal est quantifié, codé, puis transmis.
A la réception, il est décodé, puis converti en amplitude continue : après interpolation, on souhaite retrouver l'image la plus fidèle possible du signal original.
La statistique du signal doit donc être préservée : elle va influencer d'une façon essentielle la procédure de quantification.

- Dans le second cas, la loi de quantification est imposée par le système de traitement; une contrainte importante pour un système de traitement numérique consiste à commettre des erreurs de calcul qui soient négligeables vis-à-vis de l'incertitude sur le signal lui-même. Cet objectif doit être atteint malgré le caractère non stationnaire de certains signaux, tel le signal vocal par exemple.

L'erreur qui résulte de la quantification d'un signal déterministe est aussi déterministe ; ses propriétés peuvent donc, en principe, être établies par une approche déterministe.

En réalité, les signaux tels que, par exemple, le signal vocal, doivent être considérés comme étant aléatoires : la suite des erreurs de quantification est par conséquent aussi aléatoire, et l'on parle alors de bruit de quantification.

Donc, d'après la figure 2.02, une fois la matrice transformée obtenue, elle sera quantifiée pour que les valeurs possibles soient à l'intérieur d'un dictionnaire à nombre fini d'éléments. La quantification scalaire étant un cas particulier de la quantification vectorielle, on va présenter alors dans la seconde partie les concepts associés à la quantification vectorielle [10], [11].

- *Quantification scalaire uniforme*

La quantification scalaire est une forme particulière de la Quantification Vectorielle (QV), celle où la dimension des vecteurs est égal à un.

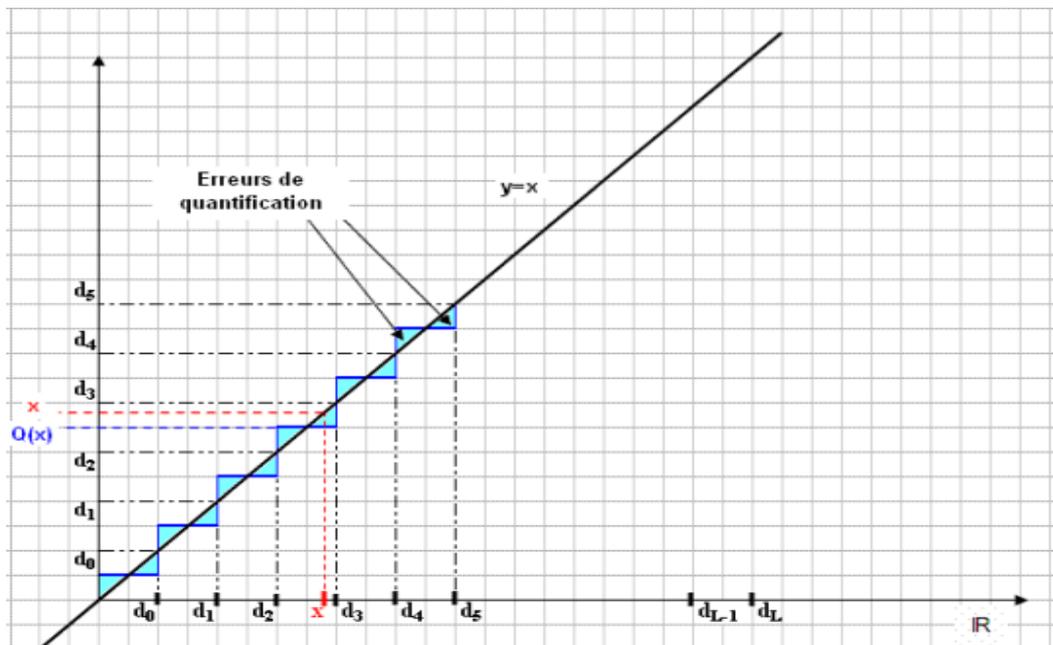


Figure 2.04: *Quantificateur uniforme à $L = 5$*

La figure 2.04 illustre la caractéristique en marche d'escalier du plus simple des quantificateurs scalaires (QS), celui uniforme à débit fixe qui est entièrement déterminé par :

- les $L + 1$ niveaux de décisions : d_0, d_1, \dots, d_L qui partitionnent en L intervalles égaux l'axe des réels \mathbb{R} et détermine le pas de quantification.
- Les L valeurs de reproduction : qui sont les centres de masses de chacun des intervalles de décision.

Cet exemple bien connu permet d'introduire les différents bruits ou erreurs de quantifications rencontrées :

- Le bruit granulaire qui se produit lorsque la valeur d'entrée x se situe dans l'une des cellules $[d_{i-1}, d_i]$, l'erreur résultante est la différence entre x et $Q(x)$. Elle peut être majorée par un demi-pas de quantification.
- Le bruit de surcharge ou dépassement qui se produit lorsque la valeur d'entrée se situe hors de l'intervalle $[d_0, d_L]$. La valeur de reproduction est alors soit y_1 soit y_L , et l'erreur résultante peut être supérieure à un demi pas de quantification.

Dans l'opération de numérisation (conversion d'une grandeur analogique vers numérique) une opération de quantification est effectuée.

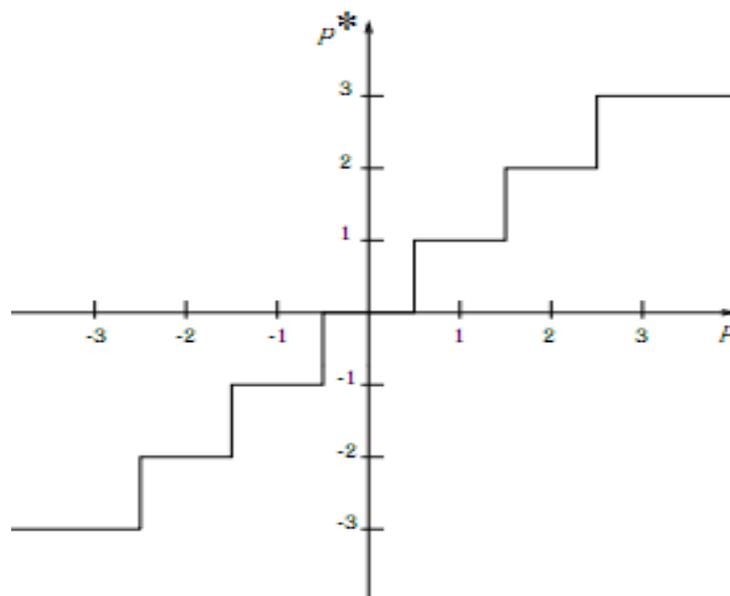


Figure 2.05: *Quantificateur uniforme à 7 niveaux et $q = 1$*

Avec: q = Pas de quantification

L'opération de quantification d'un signal p réel donne un signal p^* à valeurs discrètes en nombre fini.

Ces valeurs sont appelées *niveaux de sortie ou représentants*. Nous désignerons par quantificateur un opérateur non linéaire qui vérifie les relations entrées p – sorties p^* .

Un quantificateur scalaire uniforme avec un pas de quantification constant $q = 1$, est représenté à la figure 2.05.

Les abscisses correspondantes aux points de discontinuités de la fonction en escalier sont les *niveaux de décisions* tandis que les ordonnées correspondantes à chaque niveau de l'échelle sont les *niveaux de reconstruction*.

La différence entre les niveaux de reconstruction maximal et minimal, noté D , est appelé *gamme dynamique* du signal à quantifier.

Dans ce qui suit, nous allons décrire de façon succincte la quantification scalaire uniforme, quantification que nous utiliserons pour notre simulation en raison de sa simplicité. La fonction de quantification, notée Q est donc définie par la formule (2.01).

$$Q: \mathbb{R} \rightarrow q\mathbb{Z}$$
$$x \rightarrow mq \text{ si } x \in \left[mq - \frac{q}{2}, mq + \frac{q}{2} \right[\quad (2.01)$$

où q : pas de quantification et $m \in \mathbb{Z}$

- *Quantification vectorielle*

- La quantification consiste en l'approximation d'un signal d'amplitude continue par un signal d'amplitude discrète.
- La quantification vectorielle (QV) consiste à représenter tout vecteur x de dimension k par un autre vecteur y_i de même dimension, mais ce dernier appartenant à un ensemble fini Δ de L vecteurs.
- Les vecteurs y_i sont appelés vecteurs représentants, vecteurs de reproduction ou code-vecteurs. Δ est le dictionnaire ou le catalogue des formes.

- On représente un vecteur x de l'espace \mathcal{R}^k par une matrice colonne :
 $x(n) = (x(1), x(2), \dots, x(k))^T$
 Où $(.)^T$ indiquant l'opération de transposition d'une matrice.
- Un quantificateur vectoriel de dimension k et taille L peut être défini mathématiquement comme une application Q de \mathcal{R}^k vers Δ .

$$Q: \mathcal{R}^k \rightarrow \Delta \quad (2.02)$$

$$x \rightarrow Q(x) = y_i$$

Avec :

$$\Delta = \{y_i \in \mathcal{R}^k \mid i = 1, 2, \dots, L\}$$

La quantification vectorielle offre la combinaison des opérations d'encodage et de décodage illustrée à la figure 2.06, [12].

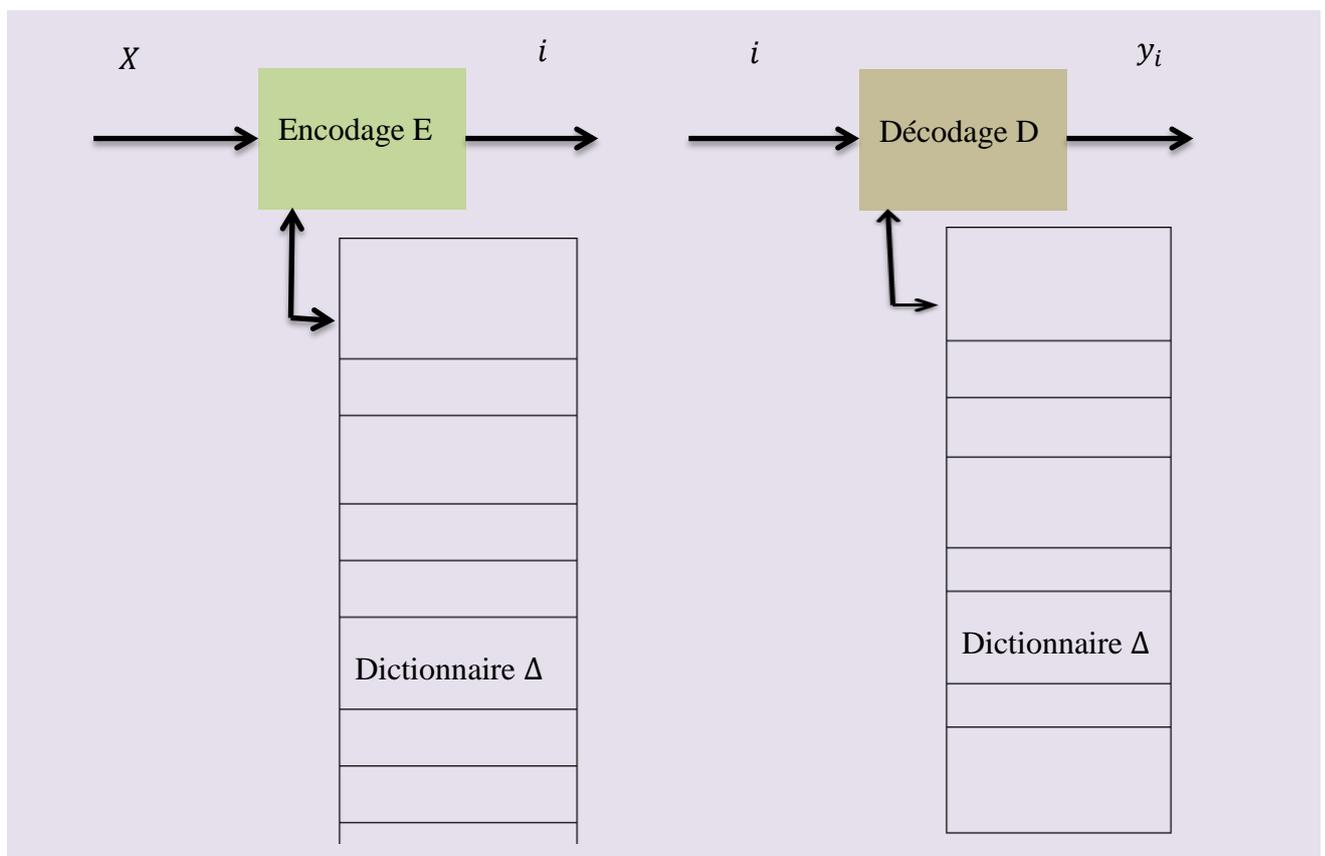


Figure 2.06: Schéma synoptique d'un quantificateur vectoriel

Décodage

Le décodeur est considéré comme un récepteur chargé de la reconstruction du signal, pour cela il doit disposer d'une copie identique du dictionnaire qu'il consulte afin de restituer le code vecteur source correspondant à l'index qu'il reçoit.

Le décodeur réalise l'opération de décompression.

2.3.2 Codage

2.3.2.1 Définitions

- On appelle « codage », une méthode permettant de passer une représentation de données vers une autre [13].

2.3.2.2 Codage entropique

Le principe du codage entropique repose sur la figure 2.07, [7].

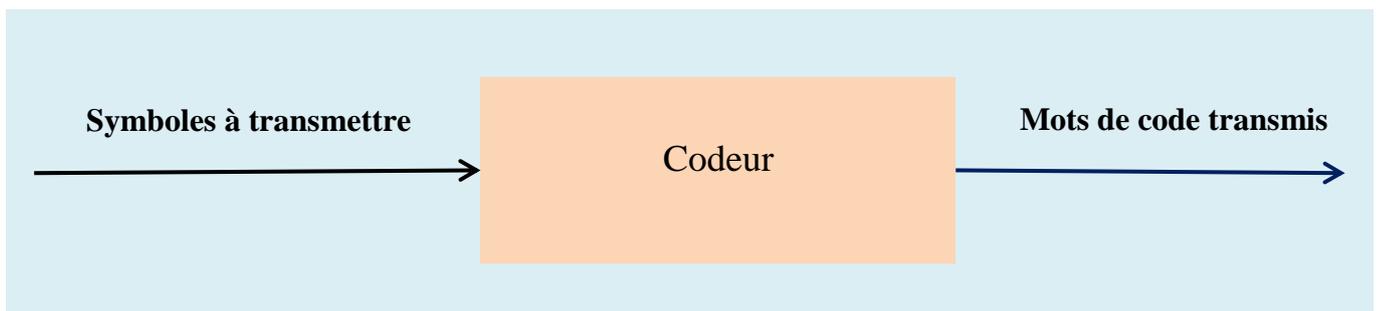


Figure 2.07: Principe du codage entropique

Un codeur entropique est illustré de la manière suivante:

- Soit un alphabet de mots de code de longueurs différentes

Exemple : 00, 01, 10, 110, 010, 111, ...

- La première étape à suivre est d'attribuer aux symboles les plus probables, les mots de code les plus courts. Ceci dans le but de réduction du coût moyen (débit en bit) de la transmission.
- Afin d'obtenir une compression efficace, l'application d'une transformation ne suffit pas, une méthode de codage sert à minimiser les redondances de la transformation.

- Parmi les algorithmes utilisés en compression d'image, figure l'algorithme de Huffman qui construit un encodeur binaire symbole par symbole optimal par rapport à la longueur moyenne des codes binaires.
- Cela veut dire que l'espérance de la longueur moyenne des codes binaires encodés par cet encodeur sera plus proche de l'entropie qu'elle l'aurait été par tout autre encodeur binaire symbole par symbole.

Exemple de codage entropique: Codage de Huffman

Entrée : Une source de symboles finie (A, P) comprenant un alphabet de symboles A et l'ensemble P des probabilités de survenir associées à chaque symbole.

Sortie :

Un encodeur binaire (aussi appelé dictionnaire de symboles) associant à chaque symbole un code binaire.

Principe :

Attribuer au symbole le plus probable le code le plus court et au symbole le moins probable le code le plus long. Cette correspondance est régie par un arbre de codage.

Codage de Huffman :

- Proposition de Huffman

On considère K symboles dont la probabilité d'occurrence est classée par ordre croissant c'est à dire $p_k < p_{k+1}$, $\{(x_1, p_1), (x_2, p_2), \dots, (x_k, p_k)\}$.

On agrège les deux symboles de probabilité minimale x_1 et x_2 en un seul symbole $x_{1,2}$ de probabilité $p_{1,2} = p_1 + p_2$.

Un arbre préfixe optimal pour les K symboles s'obtient en construisant un arbre de préfixe $k - 1$ symboles $\{(x_{1,2}, p_{1,2}), (x_3, p_3), \dots, (x_k, p_k)\}$.

Et en divisant la feuille $x_{1,2}$ en 2 nœuds fils correspondant à x_1 et x_2 .

L'algorithme de Huffman itère ce regroupement $k - 1$ fois pour reconstruire progressivement de bas en haut un arbre de code préfixe. Le code de Huffman vérifie le théorème de Shannon [16].

– L'arbre de Huffman

On appelle arbre de Huffman un arbre binaire tel que tout sous arbre a soit 0 soit 2 fils (il est localement complet).

Dans ce dernier cas, on assigne le symbole « 1 » à l'arête reliant la racine locale au fils gauche et « 0 » au fils droit.

A chaque feuille d'un arbre de Huffman, on peut associer un mot de $\{0,1\}^+$: c'est la chaîne des symboles marquant les arêtes d'un chemin depuis la racine jusqu'à la feuille.

Le maximum de l'ensemble des longueurs des mots d'un arbre de Huffman est appelé hauteur de cet arbre.

On appelle code de Huffman l'ensemble des mots correspondant aux chemins d'un arbre de Huffman ; la hauteur de cet arbre est appelée aussi hauteur du code C.

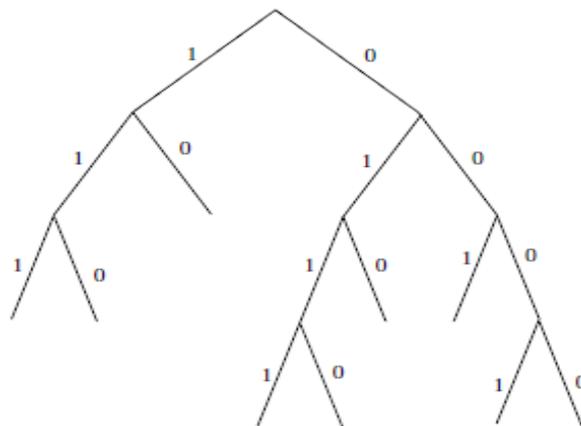


Figure 2.08: *Arbre de Huffman*

Interprétations :

Le code correspondant à l'arbre de la figure 2.08 est :

$\{111,110,10,0111,0110,010,001,0001,0000\}$, la hauteur de cet arbre est donc de 4.

– *Principe*

Les étapes nécessaires pour générer le code Huffman pour des symboles dont les probabilités d'occurrence sont connues peuvent être regroupées comme suit :

- Classer tous les symboles dans l'ordre de leur probabilité d'occurrence,

- Fusionner les deux symboles les moins probables pour former un symbole combiné, et reclasser encore les symboles dans l'ordre de probabilité : Cela génère une arborescence dont chaque nœud représente la probabilité cumulative de tous les nœuds en dessous,
- Tracer un chemin vers chaque feuille, en notant la direction sur chaque nœud.

– *Algorithme de codage de Huffman*

La source à coder est $S = (S, P)$, l'alphabet de codage V . Il est nécessaire à l'optimalité du résultat de vérifier que $|V| - 1$ divise $|S| - 1$ (afin d'obtenir un arbre localement complet).

Dans le cas contraire, il est facile de rajouter des symboles à S , de probabilités d'occurrence nulle, jusqu'à ce que $|V| - 1$ divise $|S| - 1$.

Les mots de codes associés (les plus longs) ne seront pas utilisés.

On construit avec l'alphabet source S un ensemble de nœuds isolés auxquels on associe les probabilités de P .

Soient $p_1, \dots, p_{|V|}$ les $|V|$ symboles de plus faibles probabilités. On construit un arbre (sur le modèle des arbres de Huffman) dont la racine est un nouveau nœud et auquel on associe la probabilité $p_1, \dots, p_{|V|}$ et dont les branches sont incidentes aux nœuds $p_1, \dots, p_{|V|}$. La figure suivante montre un exemple de cette itération pour $|V| = 2$.

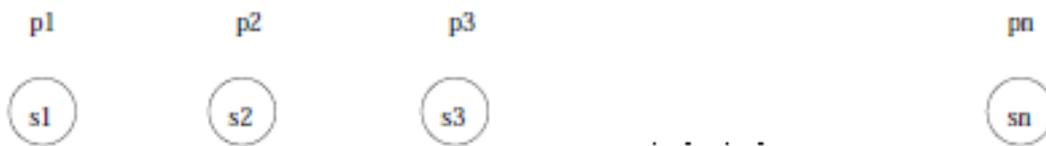


Figure 2.09: *Algorithme de Huffman de départ*

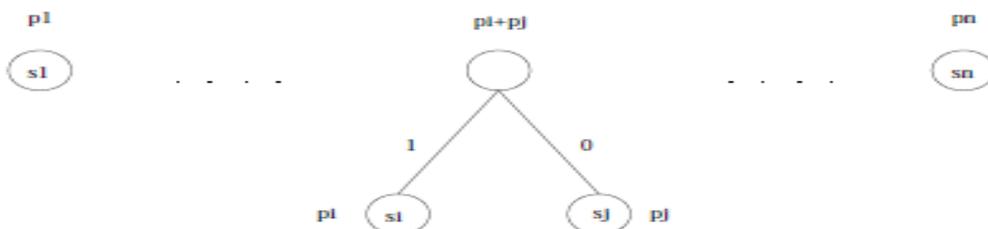


Figure 2.10: *Première étape de l'algorithme Huffman*

On recommence ensuite avec les $|V|$ plus petites valeurs parmi les nœuds du plus haut niveau (les racines), jusqu'à obtenir qu'un arbre (à chaque itération, il y a $|V| - 1$ éléments en moins parmi les nœuds de plus haut niveau), dont les mots de S sont les feuilles, et dont les mots de code associés dans le schéma ainsi construit sont les mots correspondant aux chemins de la racine aux feuilles.

Algorithme d'Encodage

- Prendre image
- Classer les symboles et leur probabilité par ordre d'occurrence croissante
- Tant que (probabilité < 1)
 - o Fusionner la probabilité des symboles les moins probables
- Fin tant que
- Tracer le chemin des nœuds vers la racine principale
- Enregistrer l'arbre construit
- Remplacer les pixels par leur code correspondant dans l'arbre
- Prendre l'image codée
- Remplacer les pixels codés par leur correspondant sur l'arbre de Huffman

2.3.2.3 Codage sans perte

On peut classer le codage en deux grandes catégories : le codage sans perte et avec perte. C'est sans doute un critère de comparaison le plus important pour les algorithmes de compression.

Une bonne partie des schémas de compressions utilisées sont appelés sans pertes, cela signifie que lorsque des données sont compressées et ensuite décompressées, l'information originale contenue dans les données a été préservée. Aucune donnée n'a été perdue ou oubliée. Les données n'ont pas été modifiées [14] [15].

Dans le codage sans perte, le taux de compression est faible (1,5 à 2).

- *Du codage source*
 - On appelle « codage source », une méthode permettant de faire de la compression de données. C'est à dire si on peut reconstruire parfaitement la source au destinataire.
 - Le but du codage de source sans pertes est de comprimer ces données de façon telle que l'on puisse reconstruire parfaitement (sans pertes, sans erreur) la source au destinataire.
 - D'autres systèmes de codage de source sans pertes ont été proposés pour prendre en compte les dépendances temporelles (d'un symbole à l'autre) de la source (source avec mémoire).
 - Ces systèmes de codage permettent de coder une source quelconque sans connaître a priori ses statistiques (codage universel), mais sont plus complexes à mettre en œuvre.
 - Outre le Codage de Huffman et de Shannon, les plus connus sont les systèmes de codage: de Lempel-Ziv et de Codage arithmétique.

2.3.2.4 Codage avec perte

La méthode de compression avec pertes jette, de façon sélective, quelques données d'une image. Ceci dans le but d'effectuer une compression avec un taux de compression meilleur.

On entend par meilleur par rapport aux méthodes de compression sans pertes.

Les algorithmes avec pertes s'appliquent généralement aux données ayant de forts taux de redondance, comme les images, ou les sons.

Certaines méthodes tirent partis d'algorithmes heuristiques élaborés qui s'ajustent eux-mêmes pour trouver le rapport de compression maximum possible en changeant aussi peu que possible les détails visibles d'une image.

Autrement, d'autres algorithmes moins élégants suppriment carrément la portion la moins significative de chaque pixel [14], [16].

Avec le codage avec perte, ou encore irréversible, les redondances sont supprimées c'est-à-dire de l'information inutiles, car ceci est invisible par le système visuel humain (SVH) et le taux de compression est élevée (> 10).

2.4 Méthodes d'évaluation d'une chaîne de compression

Evaluer les résultats obtenus sert à juger la performance du système de compression qu'on a utilisé.

Or, une simple observation basée sur la perception psychovisuelle ne suffit pas pour affirmer cette performance d'où la nécessité d'outils non subjectifs pour mesurer la performance des résultats.

L'idée fondamentale est d'utiliser soit une métrique ou une mesure; on retrouve plusieurs outils d'évaluation.

2.4.1 Outils de mesure d'évaluation

- Une métrique (satisfaisant les 3 axiomes de distance en mathématique) qui donne la distance entre deux images.
- Dans le cas de la compression, c'est la distance entre l'image originale et l'image reconstruite.
- On peut citer, la MSE (Mean square error),
- Le PSNR (Peak signal noise ratio) signifiant rapport du signal sur bruit maximum et
- Le SSIM (Structural SIMilarity) ou index de similarité structurelle.
- Ou une mesure qui donne une valeur pour une répartition des valeurs des pixels à l'intérieur d'une zone donnée de l'image.

2.4.2 Différents types de mesure d'évaluation

Avant de définir un à un et succinctement ces mesures, on va parler tout d'abord des critères de qualité concernant les techniques de codage.

Ces techniques de codages sont basées sur trois critères de qualités :

- Le taux de compression : c'est le rapport de la taille du fichier codé sur la taille du fichier initial
- La qualité du codage : sans ou avec perte (avec le pourcentage de perte)
- La vitesse de codage et de décodage

2.4.2.1 Le taux de compression

Le taux de compression donne une mesure de performance des méthodes de compression pour les images fixes.

$$Taux = \frac{\text{Nombre de bit dans l'image originale}}{\text{Nombre de bit dans l'image comprimée}} \quad (2.03)$$

Le débit de compression constitue une mesure alternative utilisé qui donne le nombre de bit nécessaire pour décrire un pixel de l'image comprimée.

$$Débit = \frac{\text{Nombre de bit par pixel dans l'image originale}}{Taux} \quad (2.04)$$

2.4.2.2 La métrique MSE

On peut trouver des mesures de distorsion ou de qualité permettant de comparer les techniques de codages. Leur utilisation est pratique car elle ne nécessite que des calculs automatiques, simples et rapides.

La mesure la plus utilisée est l'EQM (Erreur Quadratique Moyenne) ou MSE, calculée entre les pixels de l'image originale et de l'image reconstruite [17].

$$MSE = \frac{1}{N} \sum_{p \in P} (x_p - \bar{x}_p)^2 \quad (2.05)$$

Avec :

- P : Ensemble des N pixels
- p : Pixel
- N : Nombre total des pixels
- x_p : Amplitude du pixel de l'image originale
- \bar{x}_p : Amplitude du pixel de l'image reconstruite

Le choix pour cette mesure d'erreur est le plus souvent utilisé dans les traitements de données, puisqu'il est possible de développer des techniques de codages qui la minimisent. On décrit également le rapport crête du signal sur bruit ou PSNR.

2.4.2.3 Le PSNR

Le PSNR est donnée par la formule suivante :

$$PSNR = 10 \log_{10} \left(\frac{x_{max}^2}{MSE} \right) \quad (2.06)$$

Avec :

- x_{max} = Luminance maximale de l'image

Une valeur infinie du PSNR ou une valeur nulle du MSE correspond à une image parfaitement reconstruite. Le PSNR décroît en fonction de la distorsion ou de la dégradation due à la quantification.

Les mesures donnent une valeur numérique concernant la dégradation. De plus, il ne représente pas parfaitement la qualité perçue par les observations humaines, spécialement dans les taux de compression élevée [18].

2.4.2.4 La métrique RMSE

Pour deux images couleurs de dimensions $N \times M$ chacune, représentées dans le système RGB avec 8x3 bits, la métrique RMSE (Root Mean Square Error) est définie par la relation suivante :

$$RMSE(I, \tilde{I}) = \left(\frac{1}{3MN} \sum_{C \in \{R,G,B\}} \sum_{m=1}^M \sum_{n=1}^N (I_C(m,n) - \tilde{I}_C(m,n))^2 \right)^{1/2} \quad (2.07)$$

En compression, RMSE peut donner une mesure quantitative de la dissimilarité entre l'image originale et l'image reconstruite.

Inconvénient :

RMSE ne tient pas compte des propriétés de perception c'est à dire non adapté au SVH. Pour remédier à cet inconvénient, on utilise la métrique SSIM qui se complète avec le RMSE.

2.4.2.5 La métrique SSIM

La métrique SSIM a été suggérée et développée par Bovik, Wang et Simoncelli dans une série d'articles.

Elle a été développée pour mesurer la qualité visuelle d'une image compressée, par rapport à l'image originale. L'idée de SSIM est de mesurer la similarité de structure entre les deux images.

Tout d'abord, on commence par le cas des images de niveaux de gris avant d'étendre la métrique aux images couleurs.

- L'idée est de mesurer l'erreur perceptuelle comme la quantité de perte d'information structurelle, c'est-à-dire des formes dans l'image.

C'est-à-dire la similarité de structure entre les deux images, plutôt qu'une différence pixel à pixel comme le fait par exemple le PSNR.

- La SSIM prend une valeur entre 0 et 1 qui indique la corrélation par rapport à la source, 1 indiquant une corrélation parfaite.

Ainsi, on normalise *la luminance* et *les contrastes* de l'image pour ensuite regarder localement la corrélation entre les pixels de l'image.

- La métrique SSIM donne *une carte des erreurs* qui consiste en des valeurs de l'erreur locale sur toutes les zones de l'image.

Une unique valeur numérique pour l'ensemble de l'image est ensuite obtenue par une somme des erreurs locales.

Pour faciliter les calculs et la notation, on considérera les images à comparer I et I' comme des vecteurs x et y de R^N avec 0 représentant l'intensité du gris neutre.

On définit la *luminance globale* de l'image comme la moyenne de son intensité lumineuse :

$$\mu_x = \frac{1}{N} \sum_{n=1}^N x_n \quad (2.08)$$

Un changement de luminance de l'image se fait en additionnant un vecteur dont toutes les composantes sont égales.

Pour enlever l'effet du changement de la luminance, on projette l'image sur l'hyperplan :

$$\sum_{n=1}^n x_n$$

Le *contraste* d'une image est l'étendue du changement de l'intensité lumineuse de l'image par rapport à la luminance moyenne de l'image.

On le définit par l'écart-type de l'image, c'est-à-dire par la racine carrée de la variance de l'image :

$$\sigma_x = \left(\frac{1}{N} \sum_{n=1}^N (x_n - \mu_x)^2 \right)^{1/2} \quad (2.09)$$

L'effet du contraste est neutralisé en divisant l'image par son contraste. On obtient ensuite une image de luminance moyenne gris neutre et de contraste unitaire :

$$\frac{x - \mu_x}{\sigma_x} \quad (2.10)$$

Finalement, on mesure le changement structurel de l'image, c'est à dire la perte de corrélation entre $(x - \mu_x)/\sigma_x$ et $(y - \mu_y)/\sigma_y$.

On a donc deux images avec la même luminosité globale et le même contraste, tout ce qui change est la structure de l'image, c'est-à-dire sa forme, son aspect.

$$\sum_{i=1}^N x_i = 0 \quad (2.11)$$

Le changement structurel de x est représenté par une rotation du vecteur x . On mesure la corrélation entre $(x - \mu_x)/\sigma_x$ et $(y - \mu_y)/\sigma_y$ comme le cosinus de l'angle entre les deux vecteurs. On a donc :

$$\text{corr} \left(\frac{x - \mu_x}{\sigma_x}, \frac{y - \mu_y}{\sigma_y} \right) = \frac{\left\langle \frac{x - \mu_x}{\sigma_x}, \frac{y - \mu_y}{\sigma_y} \right\rangle}{\left\| \frac{x - \mu_x}{\sigma_x} \right\| \left\| \frac{y - \mu_y}{\sigma_y} \right\|} \quad (2.12)$$

Or :

$$\left\| \frac{x - \mu_x}{\sigma_x} \right\| = \left\| \frac{y - \mu_y}{\sigma_y} \right\| = \sqrt{N} \quad (2.13)$$

En écrivant :

$$\sigma_{xy} = \frac{1}{N} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (2.14)$$

On obtient :

$$\text{corr}\left(\frac{x-\mu_x}{\sigma_x}, \frac{y-\mu_y}{\sigma_y}\right) = \frac{\sigma_{xy}}{\sigma_x\sigma_y} \quad (2.15)$$

Le facteur de corrélation prend des valeurs entre -1 et 1 . La valeur 1 représente le cas où les deux images ont la même forme et la valeur -1 celui où les intensités lumineuses ont été inversées.

On introduit ensuite des petites constantes positives au numérateur et au dénominateur pour s'assurer de la stabilité de cette mesure de corrélation lorsque le dénominateur s'approche de zéro. On obtient ainsi :

$$s(x, y) = \frac{\sigma_{xy} + C}{\sigma_x\sigma_y + C} \quad (2.16)$$

Dans l'algorithme utilisé, C a été assigné arbitrairement à la valeur $(0.03 \cdot 255)^2/2$.

Il reste maintenant à évaluer la distorsion causée par le changement de luminance et de contraste de l'image.

L'erreur sur la luminance est donnée par la distorsion entre la luminance de deux images x et y :

$$I(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (2.17)$$

Où $C \ll 1$ est une petite constante définie pour éviter l'instabilité quand le dénominateur est proche de 0. La valeur $C = (0,01 \cdot 255)^2$ fait l'affaire.

D'après la loi de Weber-Frechner, $\mu_y = R \cdot \mu_x$ et en substituant dans l'équation 2.12 de la distorsion de la luminance, on obtient :

$$I(x, y) = \frac{2R + C_1 / \mu_x^2}{1 + R^2 + C_1 / \mu_x^2} \quad (2.18)$$

On néglige les termes $\frac{C_1}{\mu_x^2}$ car on suppose que C_1 est assez petit par rapport à μ_x .

On obtient ainsi que la distorsion de la luminance ne dépend pas de la luminance globale, mais seulement du ratio entre les deux luminances.

On mesure ensuite l'erreur sur le contraste comme la distorsion entre le contraste des deux images.

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (2.18)$$

Où $C_2 \ll 1$ est encore une fois définie pour éviter l'instabilité quand le dénominateur est proche de 0. Ici, on peut par exemple choisir $C_2 = (0,03 \cdot 255)^2$

On rassemble l'erreur sur la luminosité, le contraste et la structure des images, par une fonction des trois quantités :

$$S(x, y) = f(l(x, y), c(x, y), s(x, y)) \quad (2.19)$$

Il a été suggéré de prendre comme indice de similarité structurelle simplement le produit des trois quantités.

Donc finalement, la métrique SSIM calculée sur plusieurs fenêtres d'une image est la mesure entre deux fenêtres et de taille $N * N$:

$$SSIM(x, y) = l(x, y)c(x, y)s(x, y)$$

$$SSIM(x, y) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \right) \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \right) \left(\frac{\sigma_{xy} + C}{\sigma_x\sigma_y + C} \right) \quad (2.20)$$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Avec :

- μ_x = La moyenne de x
- μ_y = La moyenne de y
- σ_x^2 = La variance de x
- σ_y^2 = La variance de y

- σ_{xy} = La covariance entre x et y
- $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$: deux variables destinées à stabiliser la division quand le dénominateur est très faible
- $k_1 = 0,01$ et $k_2 = 0,03$: sont données par défaut
- L = La dynamique des valeurs des pixels, soit 255 pour des images codées sur 8 bits

2.5 Conclusion

L'objectif de la compression d'image est d'obtenir une image de taille plus petite mais fidèle à l'originale.

Parmi les techniques de compression, on trouve la compression d'images dans le domaine de transformé qui est optimale au sens de la reconstruction et réduit efficacement la dimension de l'image dans un nouveau repère.

Néanmoins, une étape de quantification et d'un codage sont indispensables pour compléter la chaîne de compression.

La notion de codage qui consiste à réduire la taille de l'image afin de n'occuper qu'une moindre place pendant leur stockage, les types de codage ne conservant tous les informations et celles qui ne gardent qu'une partie des données n'ont pas été présentés ici mais sont donnés en annexe deux.

CHAPITRE 3

MODELE DE COMPRESSION D'IMAGE AVEC TKL

3.1 Introduction

Le but de la compression d'image est de ranger l'image sous une autre forme d'image plus compacte. C'est à dire une représentation qui ne requiert que quelques bits pour l'encodage que l'image originale.

Ceci est possible pour l'image parce que dans leur forme brute, elle contient un haut degré de données redondantes.

Chaque image contient plusieurs formes et structures. De ce fait, il y a plusieurs corrélations entre chaque pixel voisin.

Si on arrive à trouver une transformation réversible qui enlève cette redondance en décorrélant les données, alors une image peut être archivée efficacement. La TKL est une transformation linéaire qui pourrait accomplir cette tâche.

3.1.1 Historique

La Décomposition de Karhunen-Loeve (Karhunen, 1946; Loeve, 1955) est connue en statistique comme étant une méthode d'analyse exploratoire ou factorielle des données permettant de faire une approximation d'un système de dimension élevée par un autre de dimension nettement plus faible (*réduction de dimension*).

Il s'agit d'une application linéaire qui transforme une base orthonormée en une autre, formée des vecteurs propres (*transformation unitaire*).

Dans cette nouvelle base, les informations pertinentes peuvent mieux être classées par ordre décroissant selon leurs variances et mises en évidence à travers les projections sur les vecteurs propres (*directions de variance maximale et décorrélées*).

Ainsi, un modèle de données peut être élaboré en prenant un nombre assez réduit de projections pour représenter une proportion donnée de la variance totale (*approximations par les projections*).

Cette technique est aussi connue dans d'autres domaines scientifiques sous le nom :

- d'analyse d'Hotelling (Hotelling, 1933) ou

- de Décomposition Orthogonale aux valeurs Propres (Proper Orthogonal Decomposition (POD)) en compression de données (Andrews et al. en 1967), ou
- d'Analyse en Composantes Principales ACP (Jolliffe, 1986).

Elle présente une similarité avec la décomposition en Valeurs Singulières (SVD : Singular Values Decomposition par Golub et Van Loan en 1990) en mettant en évidence les propriétés d'une matrice en un produit de matrices plus simples et dont les caractéristiques sont clairement identifiables et interprétables.

Le point central de toutes ces applications est l'extraction des caractères dominants d'un ensemble de données, permettant ainsi d'accéder à une réduction de modèle.

3.1.2 Transformation de Karhunen-Loeve

3.1.2.1 Définitions

- On appelle *TKL*, une méthode d'analyse qui se propose de fournir des représentations synthétiques de vaste ensemble de valeurs numériques.

En traitement d'images, le modèle permet de tenir compte de l'information colorimétrique lors du traitement [19].

- La réduction de dimensionnalité est basée sur la notion de projection : régressions linéaires.
- C'est un outil pour résoudre de nombreux problèmes d'optimisation. Le principe de base, c'est de représenter des données multivariées par projection sur des plans optimaux.

Ce principe est unificateur et dérive d'un modèle mathématique unique [20].

3.1.2.2 Principe générale

a) Analyse d'image

Définition

On appelle *analyse d'une image*, une méthode qui consiste à déterminer les corrélations entre les pixels de l'image sachant qu'intuitivement, il existe des « relations cachées entre les pixels voisins ».

Ces corrélations, données par la matrice des corrélations sont utilisées pour mesurer la redondance interpixel de l'image.

Exemple :

La compression utilise ces redondances pour diminuer le nombre de bits nécessaires pour représenter un pixel. Pour cela, nous allons distinguer deux cas :

Cas 1 : les pixels se trouvent repartis à l'intérieur des trois plans de couleur R, G, B . On parle dans ce cas de corrélations ou redondances *inter-composantes*.

Cas 2 : les pixels se trouvent à l'intérieur d'un même plan de couleur. On parle dans ce cas de corrélations ou redondances *intra-composantes*.

b) *Modélisation d'une image par TKL*

Définition

On appelle *modélisation d'une image par TKL*, la décomposition de l'image dans un espace de représentation de dimension (nombre de vecteur-variable) où ses coefficients appelés "coefficients de Karhunen Loeve" seront totalement décorrélés. C'est-à-dire, de *corrélations nulles*.

Ainsi, les redondances seront éliminées et nous pouvons choisir un nombre *réduit* de *coefficients* sur lesquelles l'image sera projetée. La projection de l'image peut être prise comme étant un modèle de l'image relativement représentatif de l'image originale selon les critères et les enjeux que l'on se fixe dans la quantification.

3.1.2.3 Formulation mathématique

D'une manière générale [21], [22], [23] pour un vecteur aléatoire X de dimension n , notons ses composantes par :

$$X = \begin{pmatrix} X_0 \\ X_1 \\ \vdots \\ X_{n-1} \end{pmatrix} \quad (3.01)$$

Sa moyenne est définie par :

$$m_X = \begin{pmatrix} E(X_0) \\ E(X_1) \\ \vdots \\ E(X_{n-1}) \end{pmatrix} \quad (3.02)$$

Où $E(X_j)$ est l'espérance mathématique de X_j pour j variant de 0 à $n - 1$.

Sa matrice de covariance s'écrit :

$$\Gamma_X = \begin{pmatrix} \sigma_{00} & \sigma_{01} & \cdots & \sigma_{0,n-1} \\ \sigma_{10} & \sigma_{11} & \cdots & \sigma_{1,n-1} \\ & & \vdots & \\ \sigma_{n-1,0} & \sigma_{n-1,1} & \cdots & \sigma_{n-1,n-1} \end{pmatrix} \quad (3.03)$$

Où $\sigma_{ij} = E((X_i - m_i)(X_j - m_j)), 0 \leq i, j \leq n - 1$

Comme la matrice Γ_X est semi-définie positive, alors ses valeurs propres sont réels positifs ou nuls et il existe une matrice orthogonale Q telle que :

$$\Gamma_X = Q \begin{pmatrix} \sigma_0^2 & & 0 \\ 0 & \ddots & 0 \\ 0 & & \sigma_{n-1}^2 \end{pmatrix} Q^t \quad (3.04)$$

Où $Q \in R^{nm}$:

(i) $\sigma_0^2 \geq \sigma_1^2 \geq \cdots \geq \sigma_{n-1}^2 \geq 0$ sont les valeurs propres de Γ_X .

(ii) les vecteurs colonnes q_0, q_1, \dots, q_{n-1} de Q forment une base orthonormale de R^n , composée de vecteurs propres de Γ_X .

Le vecteur transformé de X par TKL est le vecteur Z donné par :

$$Z = \begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ z_{n-1} \end{pmatrix} = Q^t \cdot (X - m_X) \quad (3.05)$$

$$Z = Q^t \begin{pmatrix} X_0 - m_0 \\ X_1 - m_1 \\ \vdots \\ X_{n-1} - m_{n-1} \end{pmatrix} \quad (3.06)$$

$X - m_X$, est une variable centrée.

Ainsi, tout vecteur aléatoire X , de dimension n peut-être décomposé dans les sous espaces propres associés aux valeurs propres de Γ_X ;

(i) $\|X - QZ\|$ est minimum au sens de l'erreur quadratique moyenne. Cette propriété fait que la transformation de KL compacte au maximum l'énergie sur un minimum de coefficients [24].

(ii) Les composantes de Z sont totalement décorrélés et on a : $\forall i \neq j \sigma_{ij} = 0$

3.1.3 Catégories de TKL

La transformation de Karhunen peut être présentée selon deux approches (cf. § 3.1.2.2 a):

- Méthode vectorielle
- Méthode marginale

3.1.3.1 Méthode vectorielle

- Prenons une image et notons cette image I .
- La dimension de la matrice couleur est de $512 \times 512 \times 3$. Elle est associée dans le système (R, G, B) dont I (IR, IG, IB) sont respectivement les composantes couleurs pour chaque canal. L'image transformée est notée de même dimension que I .
- Trois blocs correspondants dans les trois canaux de couleurs, sont pris simultanément pour être analysés ensemble du point de vue corrélation inter-composantes.
- Chaque couleur constitue le caractère ou vecteur-variable ou variable d'analyse et les trois valeurs de pixels de même position constituent un vecteur-individu.
- Les corrélations entre les composantes couleurs sont données par la matrice des corrélations.

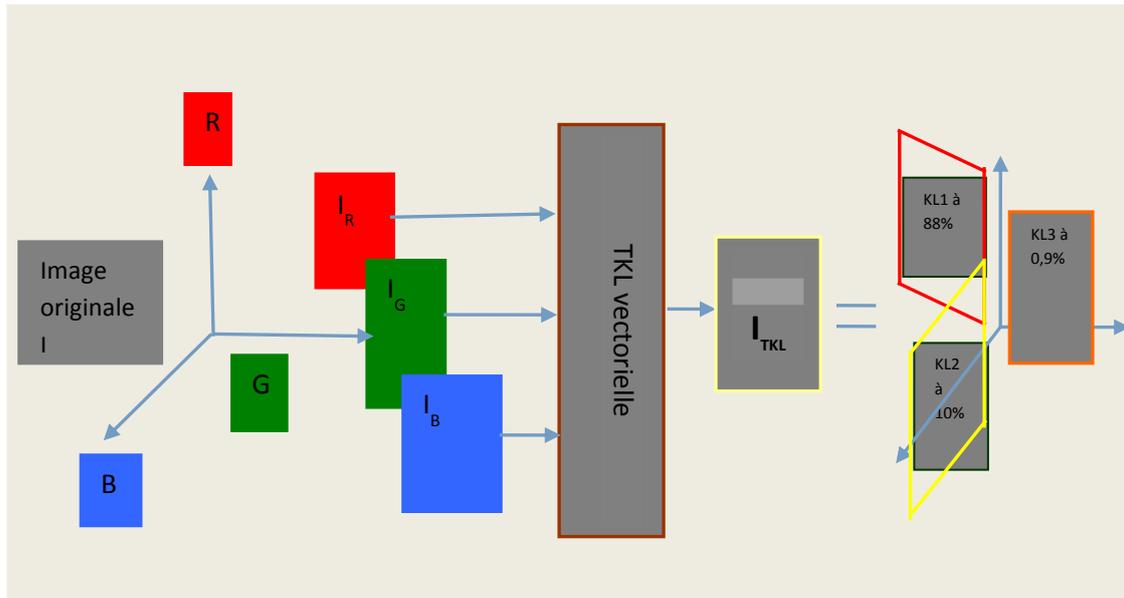


Figure 3.01: *Approche de la méthode TKL vectorielle*

3.1.3.2 Méthode marginale

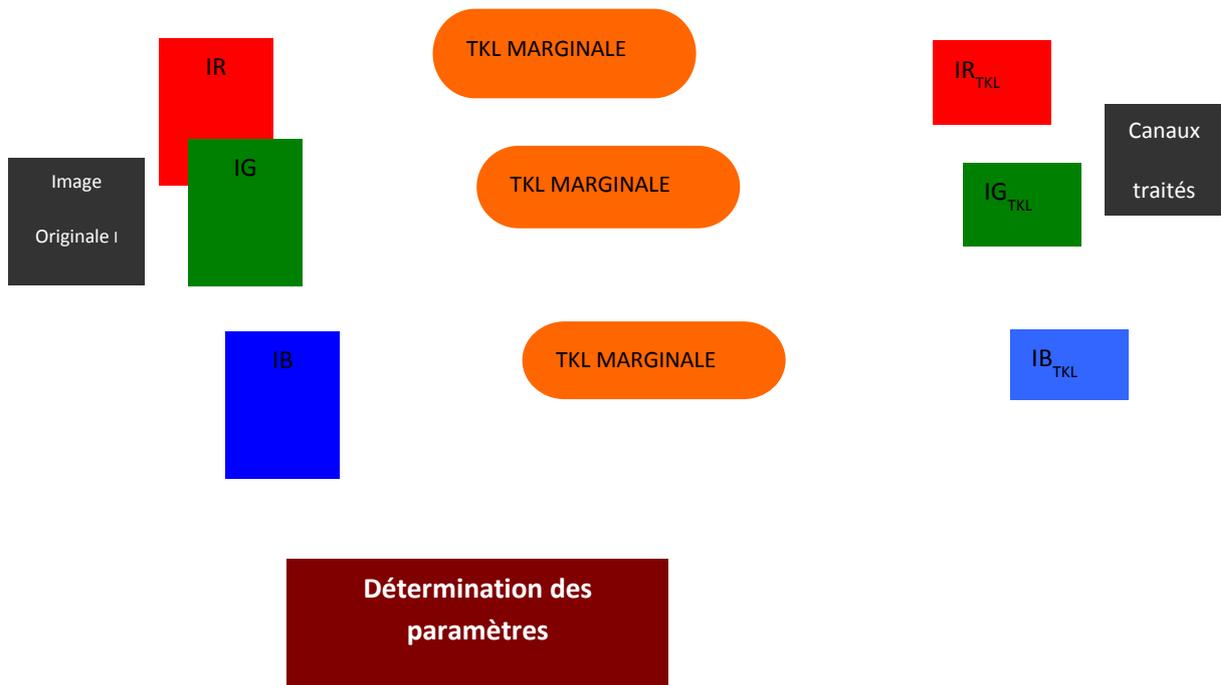


Figure 3.02: *Approche de la méthode TKL marginale*

Selon cette approche, les composantes couleurs sont prises séparément, indépendamment les unes des autres, pour être analysées du point de vue corrélations intra-composantes. Les trois canaux traités $IR_{TKL}, IG_{TKL}, IB_{TKL}$, ont des lignes ou colonnes totalement décorréliées.

3.1.4 Calcul des coefficients de TKL

Le problème consiste à dissocier puis stocker les composantes RVB (rouge, vert, bleu) de l'image dans les premiers, seconds et troisièmes coefficients de Karhunen.

- C'est l'analyse de la structure de la matrice variance-covariance : c'est-à-dire de la variabilité, dispersion des données [7], [8], [20], [24].

Le calcul de la TKL est typiquement exécuté en trouvant les vecteurs propres de la matrice de covariance, soit donc une estimation de la matrice de covariance. C'est-à-dire qu'il faut calculer les vecteurs propres correspondant aux valeurs propres.

$$Y = V X \quad (3.07)$$

Avec :

- Y : vecteur transformé
- X : vecteur de donnée
- V : la matrice de transformation

Et notons :

$$D = cov(Y) \quad (3.08)$$

$$C = cov(X) \quad (3.09)$$

$$D = [V]^T [C] [V] \quad (3.10)$$

Avec :

- D : matrice diagonale, elle contient les valeurs propres dans les diagonales et les colonnes de V sont les vecteurs de KLT.
- V : $[P_1][P_2][P_3]$ Où; P_i : Rotation dans un plan.
- C : matrice de covariances

3.1.4.1 Coefficients de Karhunen-Loeve

Les coefficients de Karhunen Loeve sont : $KL_1, KL_2, \dots, KL_k, \dots, KL_q$.

Avec :

- KL_k : la nouvelle variable qui est une combinaison linéaire d'origine X_1, \dots, X_2 ;
- $KL_k = a_{1k} X_1 + a_{2k} X_2 + \dots + a_{pk}$ sont les coefficients a_{jk} à déterminer. Ce sont les composantes des vecteurs propres, tel que KL_k soient ci-dessous :
 - o 2 à 2 non corrélés
 - o De variances maximales
 - o D'importances décroissantes

Le premier coefficient contient la plus grande quantité de variations. Le second coefficient contient en deuxième lieu la plus grande quantité de variations. Les données sont très corrélées ; si on enlève 50% des dimensions, le taux de « perte d'infos » peut être moins que 10%.

3.1.4.2 Calcul des variables et espaces d'étude

Une série à variables individuelles ; soit l'étude de la variable X , une série de valeurs définies dans \mathbb{R} ; on a :

- La moyenne

$$\overline{X(l,j)} = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M x_i \quad (3.11)$$

- Variance

$$\theta^2 = V(X, Y) = \sum_{i=1}^N \sum_{j=1}^M (x_i - \bar{X})^2 (y_i - \bar{Y})^2 \quad (3.12)$$

- L'écart type

$$\theta = \sqrt{V(X, Y)} \quad (3.13)$$

On a deux séries de valeurs définies dans \mathbb{R} : la covariance et la corrélation ;

- La covariance

$$\theta_{xy} = cov(XY) + \sum_{i=1}^N (x_i - \bar{X})(y_i - \bar{Y}) \quad (3.14)$$

- La corrélation

$$cor(X, Y) = \frac{\theta_{xy}}{\theta_x - \theta_y} = \frac{cov(XY)}{\sqrt{cov(X)}\sqrt{cov(Y)}} \quad (3.15)$$

- Matrice de covariances

Cela revient à calculer la matrice centrée, une matrice carrée, une relation affine

$$P = \begin{bmatrix} var(R) & cov(R, G) & cov(R, B) \\ cov(G, R) & var(G) & cov(G, B) \\ cov(B, R) & cov(B, G) & var(B) \end{bmatrix} \quad (3.16)$$

3.1.4.3 Recherche et décomposition en vecteurs propres et valeurs propres

- Matrice des vecteurs propres

$$V = \begin{bmatrix} 0,614 & 0,588 & 0,526 \\ -0,581 & -0,114 & 0,806 \\ 0,5346 & -0,801 & 0,271 \end{bmatrix} \quad (3.17)$$

- Matrice des valeurs propres

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \quad (3.18)$$

- Chaque image est en couleur : rouge, vert, bleu (RVB). Chaque pixel, chaque site (x, y) contient une information couleur sur l'intensité du rouge, du vert, du bleu.

Il est donc possible de diviser par trois, la taille d'une image, en ne conservant qu'un seul canal ($((RGB) \rightarrow (KL1), (x, y, z) \rightarrow x')$).

- On obtient d'une part, une réduction de l'espace couleur et d'autre part la conservation de l'axe permet de conserver le plus d'information, d'expliquer 90% de l'information.
- Le nouvel espace est une combinaison linéaire de l'espace d'origine.

$$\frac{\lambda_1}{\sum_{j=1}^3 \lambda_j} > T(0,90) \quad (3.19)$$

- On définit algébriquement, sur un échantillon de « n » observations sur un vecteur de « p » variables.

$$X = x_1, x_2, \dots, x_p \quad (3.20)$$

- On définit le premier coefficient de l'échantillon par transformation linéaire:

$$Z_1 = a_1^T X = \sum_{i=1}^p a_{i1} x_i \quad (3.21)$$

$$a_1 = (a_{11}, a_{12}, \dots, a_{p1}) \quad (3.22)$$

$$\text{var} [Z_1] \text{ Maximum} \quad (3.23)$$

Le $k^{\text{ième}}$ coefficient se déduit par transformation linéaire.

$$Z_k = a_k^T X \quad k = 1, \dots, p \quad (3.24)$$

Où le vecteur a_k (coefficients vectoriels) :

$$a_k = (a_{1k}, a_{2k}, \dots, a_{pk}) \quad (3.25)$$

Est choisi tel que :

$$\text{var} [Z_k] \text{ Maximum} \quad (3.26)$$

Et est sujet à :

$$\text{cov}[Z_k, Z_l] = 0 \text{ pour } k > l \geq 1 \text{ et à } a_k^T a_k = 1 \quad (3.27)$$

Pour la matrice de covariance P , on a :

$$P = A^T S A \quad (3.28)$$

$$P_{ij} = \lambda_i \delta_{ij} \quad (3.29)$$

- P_{ij} : la matrice de covariance des coefficients
- a_k vecteur propre de S
- La k -ième valeur propre (la plus grande) est la variance du k -ième coefficient.

3.1.4.4 Propriétés de la matrice des covariances

Si on applique ces théories à une matrice. Ainsi, l'espérance d'une matrice est la matrice des espérances de chacune de ses colonnes. On a la propriété suivante pour exprimer que l'espérance d'une matrice est linéaire [25].

- Linéarité de l'espérance d'une matrice

Soit $X_{n \times p}$ une matrice formée par p vecteurs à n composantes et soient $A_{m \times n}$ et $B_{p \times q}$ deux transformations linéaires, alors :

$$E[AXB] = AE[X]B \quad (3.30)$$

La notion de covariance se généralise de la même manière en calculant deux à deux les covariances entre les colonnes.

Les résultats seront disposés dans une matrice notée R_X , appelée *matrice des covariances*, de telle sorte qu'elle soit symétrique. On a alors les définitions suivantes

- D'une manière formelle, R_X se définit par:

$$R_X = E[(X - E[X])(X - E[X])'] = E[XX'] - E[X](E[X])' \quad (3.31)$$

On dit que X ne possède aucune *autocorrélation* ou ses colonnes sont *décorrélées* si sa matrice de covariance est diagonale.

Comme l'objectif est d'annuler les autocorrélations qui sont principalement les sources de redondances dans une image, alors il est indispensable de rechercher une transformation, si possible linéaire, qui réalise une autocorrélation nulle sur l'image.

La proposition suivante permet de trouver la solution à ce problème.

- Pour la matrice X ci-dessus, la transformation linéaire K qui décorrèle complètement X est celle dont chaque ligne est un vecteur propre de la matrice de covariance R_X . Les vecteurs propres étant ordonnés selon l'ordre décroissant des valeurs propres correspondantes (sous forme matricielle, voir paragraphe 3.1.4).

Soit :

$$Y = KX \quad (3.32)$$

Nous avons:

$$R_Y = E[YY'] - E[Y](E[Y])' \quad (3.33)$$

$$R_Y = E[(KX)(KX)'] - E[KX](E[KX])' \quad (3.34)$$

$$R_Y = E[KXX'K'] - (KE[X])(KE[X])' \quad (3.35)$$

$$R_Y = KE[XX']K' - KE[X]E[X]'K' \quad (3.36)$$

$$R_Y = K(E[XX'] - E[X]E[X]')K' \quad (3.37)$$

$$R_Y = KR_XK' \quad (3.38)$$

Comme R_X est symétrique et réelle pour une image, alors elle est diagonalisable par une matrice orthogonale.

C'est grâce à cette propriété que R_Y est aussi diagonalisable.

K est alors une matrice de passage orthogonale qui diagonalise R_X . La transformation associée à K est appelée transformation de Karhunen-Loeve.

3.1.4.5 Décorrélation des données

Les données provenant des pixels voisins sont fortement corrélées pour plusieurs images; généralement, on utilise une image de 512*512 pixels, dans la taille avec chaque niveau de gris de canal de couleur de pixels sont représentés par huit bits pour une gamme de [0,255], [26].

3.2 Modèle de compression par TKL

3.2.1 Chaîne de compression avec TKL

- Objectifs :

Le but de la compression utilisant la TKL c'est de garder un maximum d'informations des données initiales. C'est à dire, décrire à l'aide de quelques coefficients un maximum de cette variabilité. Ce qui permet :

- une réduction des données à q nouveaux descripteurs.
- une visualisation des données à 2 ou 3 dimensions

On verra dans la figure 3.03 la chaîne de compression utilisant cette transformation.

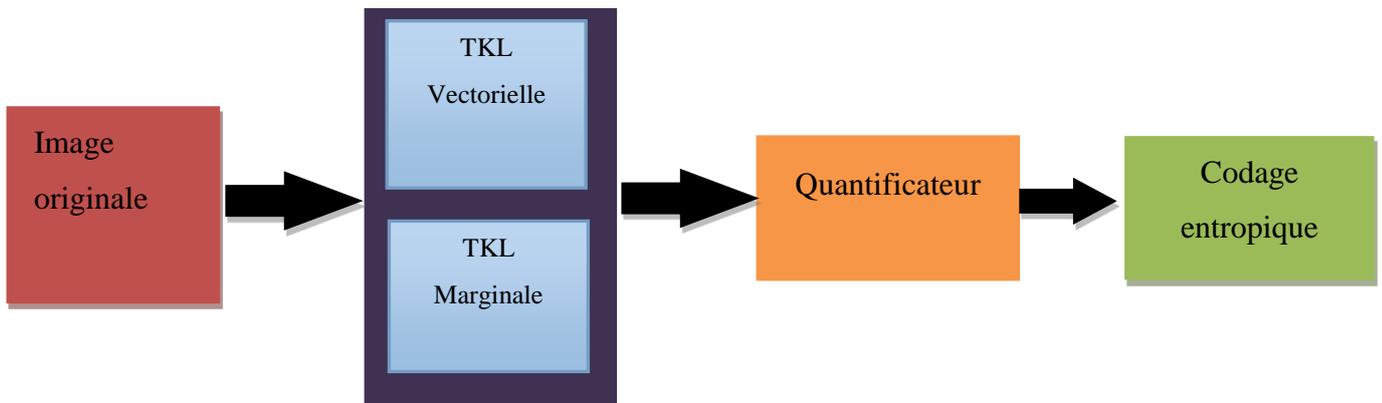


Figure 3.03: Compression d'image par TKL

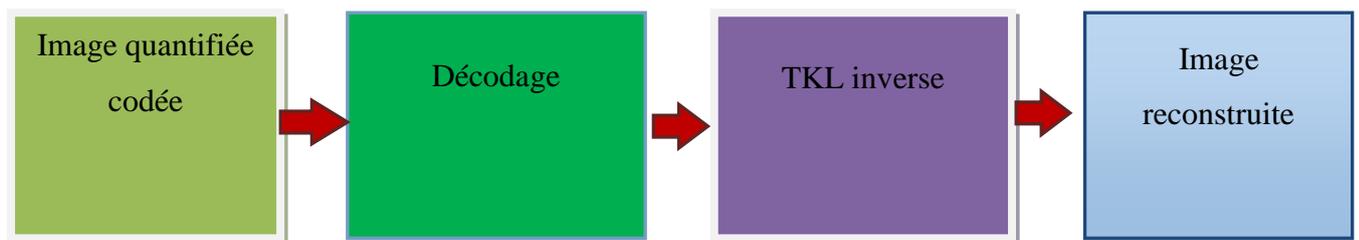


Figure 3.04: Décompression d'image par TKL

3.2.2 Algorithme de la compression avec TKL

3.2.2.1 Etapes de la mise en œuvre de la TKL

Pour mettre en œuvre la TKL tout d'abord, il faut passer par les étapes suivantes :

- ❖ Préparation de la matrice des données :

Disposer les données en colonnes et en lignes pour qu'elles forment une matrice dont les colonnes représentent les caractères ou variables et les lignes les individus.

- ❖ Calculs des paramètres statistiques de la matrice :

- ❖ Par définition, ces paramètres sont calculés pour chaque colonne. Ce sont les paramètres de position et de dispersion, qui caractérisent la distribution.
- ❖ Calcul ou estimation de la matrice covariance
- ❖ Calcul des valeurs propres de la matrice covariance, triage décroissant des valeurs propres et formation de la matrice des vecteurs propres associés aux valeurs propres triées.
- ❖ Calcul de la matrice transformée
- ❖ L'application de ces étapes pour une image couleur donnée et selon une approche donnée, peut être résumée par l'algorithme illustrée par la figure (3.03), [7], [8], [26], [27].

3.2.2.2 Algorithme de calcul de la TKL

- Charger l'image
- Décomposer en trois matrices représentant les trois composantes de l'image
- Réaliser un tableau de données :
 - considerer les images comme des vecteurs de
 - $(img(i, j) = v((i-1) * (p+j)))$.
- Faire la moyenne de chaque vecteur
- Calculer les valeurs propres et les vecteurs propres
- Calcul de la matrice transformée

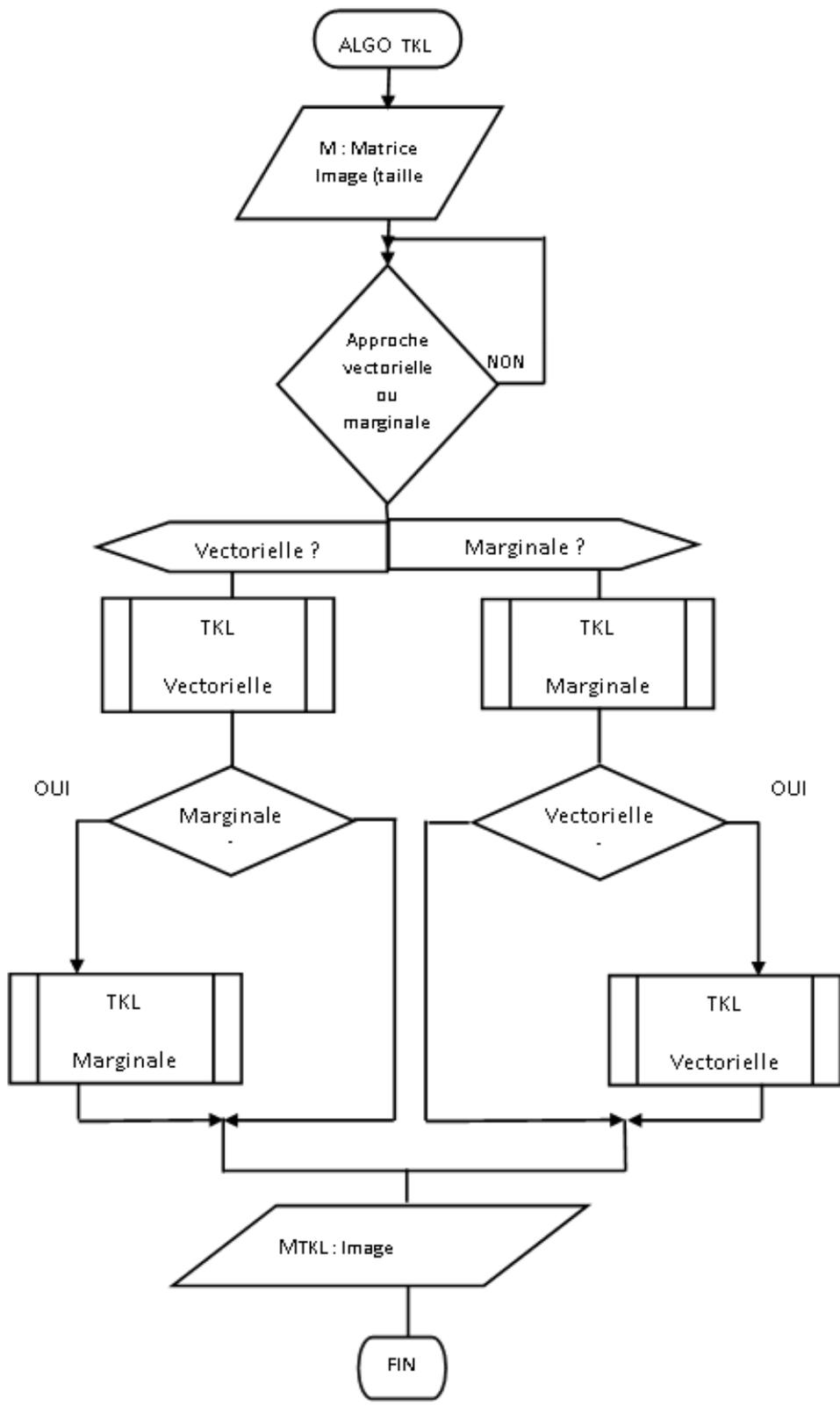


Figure 3.05: *Etapes de mise en œuvre de la TKL*

Remarques :

L'énergie d'un axe est le ratio de la valeur propre de l'axe sur la somme des valeurs propres. Cette énergie est la quantité d'information expliquée par cet axe.

La qualité de la reconstruction en gardant les deux axes les plus importants est simplement la somme des énergies de ces deux axes.

Plus la qualité sera proche de 100%, plus la reconstruction sera bonne, ce qui est équivalent à dire que le dernier axe ne contient que très peu d'informations.

3.2.2.3 Algorithme de compression avec TKL

- {
- Prendre l'image
- Transformer l'image en coefficients de TKL
- Quantifier la matrice de la transformée
- Passer par une étape d'un codage entropique
- }

3.2.2.4 Algorithme de la décompression avec TKL

- {
- Prendre l'image quantifiée codée
- Décodage de l'image
- Appliquer la TKL inverse
- Obtenir une image reconstruite
- }

3.3 La reconstruction de l'image issue de compression par TKL

Même si la transformation de Karhunen-Loeve est optimale du point de vue décorrélation, elle ne permet directement aucune compression.

Cependant, la transformée $Y = KX$ obtenue se prête très bien à l'encodage entropique.

On pourrait même tronquer Y pour effectuer une compression plus forte et, en théorie, compresser efficacement par un encodage de Huffman par exemple.

Sous forme matricielle, l'opération de troncature peut s'écrire comme suit :

$$Y' = I_k \cdot Y \quad (3.39)$$

Où I_k est la matrice diagonale dont les k premiers éléments de la diagonale sont des 1 et le reste des 0.

On dit que : Y' est la matrice tronquée à l'ordre k de Y . Ensuite, on reconstruit X par une approximation en multipliant Y' par une matrice V . On obtient finalement une approximation de X qui est :

$$X' = V \cdot Y' \quad (3.40)$$

Comment peut-on chercher une bonne approximation de X et jusqu'à quel ordre peut-on tronquer pour que l'image après reconstruction reste encore d'une bonne qualité visuelle ?

On propose une reconstruction optimale vis-à-vis de l'erreur quadratique moyenne (E.Q.M) pour essayer à répondre à cette question en estimant l'erreur commise.

L'E.Q.M entre X et X' est minimale quand K est la transformation de Karhunen-Loeve et V est l'inverse de cette transformation c'est-à-dire :

$$V = K^{-1} \quad (3.41)$$

3.4 Conclusion

La TKL est définie comme une transformation linéaire dont les vecteurs de base sont les vecteurs propres de la matrice de covariance de l'image.

Cette transformation diagonalise la matrice de covariance, elle décorrèle les données. Les coefficients résultant de la TKL peuvent être encodés avec un algorithme entropique présentant peu de bits.

Cette transformation optimale en terme de minimisation du taux d'erreur. L'utilisation des vecteurs propres comme de vecteurs de base assure que la variance du premier coefficient est maximale et est sujette à l'orthogonalité de vecteurs de base. Les variances de coefficients sont consécutivement dans l'ordre décroissant.

La transformation est donc performante pour la technique de compression d'image. Pour un taux de compression modéré, une petite distorsion est visible.

Plus le taux de compression augmente, plus la distorsion devienne évidente.

CHAPITRE 4

SIMULATION ET EVALUATION DE LA METHODE DE COMPRESSION D'IMAGES AVEC TKLH

4.1 Introduction

La transformation de Karhunen-Loeve est une technique qui est utile pour le traitement d'image, comme l'extraction de signatures, la segmentation, la classification des données.

Dans ce chapitre, on l'utilise pour une technique de compression [28]. Le problème consiste à réduire la dimensionnalité d'un ensemble de variables plus petit que l'ensemble original de variables, qui néanmoins contient la plupart de l'information de l'échantillon.

Par information, on veut parler de la variation présente dans l'échantillon et donnée par les corrélations entre les variables originales.

4.2 Modèle de compression proposée avec la méthode TKL

On a vu dans les chapitres précédents les techniques et méthodes pour comprimer une image répondant aux exigences d'un système de compression en général et d'un autre côté, pour un système de compression utilisant la transformée.

Le problème réside sur le fait que l'image occupe une place prépondérante dans le stockage ou dans les archives :

Donc, il est nécessaire de trouver une méthode codage ou un outil mathématique assez puissant pour obtenir un taux de compression acceptable et en même temps une qualité de l'image reconstruite assez similaire à l'originale.

La modélisation mathématique proposée afin de rendre efficace le système de compression avec une transformation est la méthode avec la transformation de Karhunen Loeve qui est optimale.

La solution proposée pour la problématique citée dans l'introduction : « *Même s'il existe aujourd'hui des mémoires de stockage de masse de grande capacité, le besoin de stocker s'avère encore insatisfait* » ; Une des solutions à ce problème est le recours à des techniques de compression d'images fixes telles que Jpeg, Jpeg 2000. Mais dans notre travail, on propose une *Compression d'image* dans le domaine des transformés utilisant l'Analyse factorielle de données, notamment la **TKL**.

La transformée de Karhunen est une méthode optimale de codage des images au sens de l'erreur quadratique moyenne (EQM).

On ne trouve dans la littérature qu'une méthode adaptative dans le domaine spatiale. De nombreuses autres méthodes de codage sous-optimales ont été développées pour éviter les problèmes liés à la transformée avec l'analyse factorielle de données.

De telles méthodes ont des performances inférieures par rapport à analyse factorielle de données aussi bien du point de vue de la qualité visuelle que de l'erreur quadratique moyenne, bien que certaines d'entre elles soient tout à fait efficaces sur de processus markoviens du premier ordre.

4.3 Objectifs

La présente simulation se propose de :

- Faire une optimisation en amont d'une chaîne de traitement en utilisant la décomposition par TKL afin d'obtenir une décorrélation optimale.
- Tester et analyser, dans le cadre de la compression par TKL, les effets d'un partitionnement en blocs carrés et du choix du nombre de coefficients à retenir.
- Tester et évaluer les performances d'une chaîne de compression par TKL intégrant un quantificateur optimal et d'un codeur entropique.
- Par ailleurs, cela permet de tester l'adéquation et la bonne intégration de ces méthodes de quantification et de codage dans la chaîne.

4.4 Modèle de simulation

On représente notre simulation de la méthode de compression sous matlab suivant les modèles figurés dans les synoptiques (4.01), (4.02) suivants :

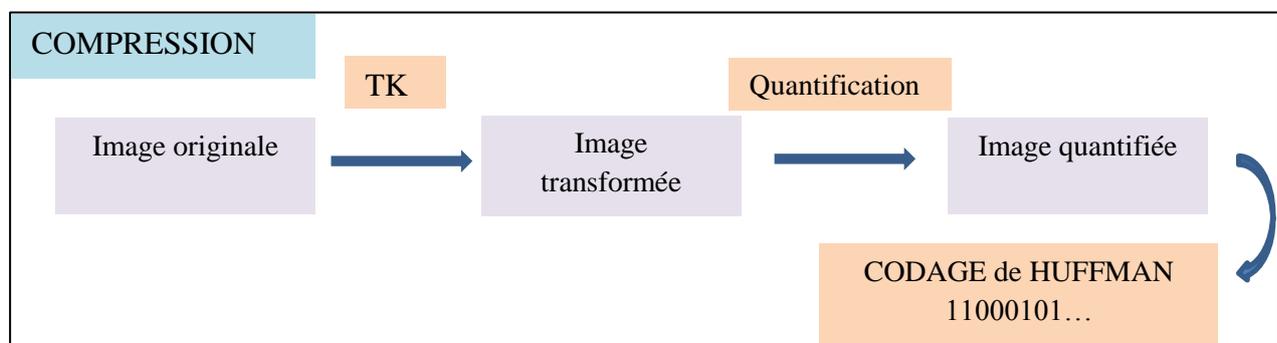


Figure 4.01: Schéma de compression avec TKL

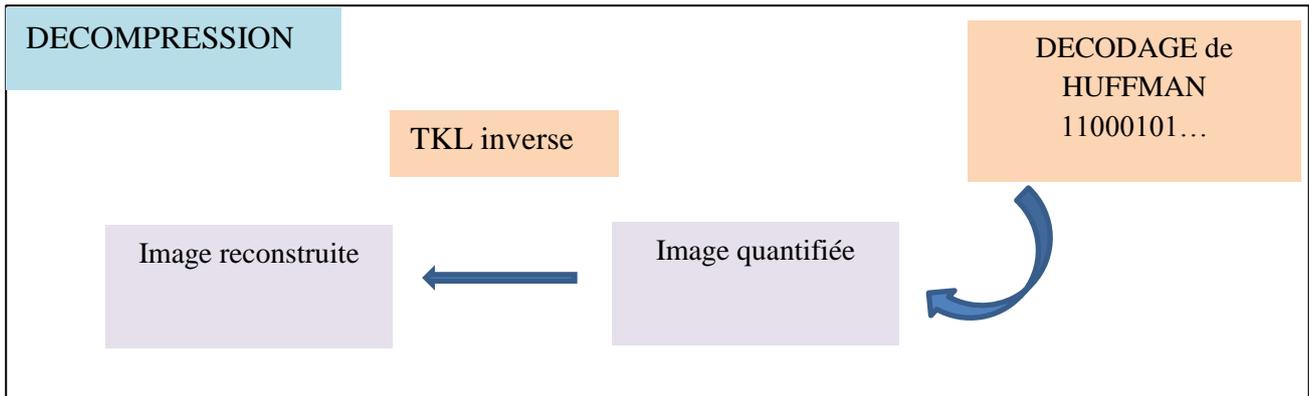


Figure 4.02: Schéma de décompression avec TKL

4.5 Résultats et évaluation de la simulation

– *Du point de vue information (énergie)*

Selon l'approche vectorielle, on applique la transformation d'analyse factorielle de données à une matrice formée de trois blocs R, G et B.

Après on projette sur les nouveaux axes factoriels avec les pourcentages d'inerties pour chaque axe de coefficient.

Dans notre cas, chaque bloc est centré, c'est-à-dire on utilise la transformation de Karhunen-Loeve (T.K.L).

Pour notre simulation, on a choisi quelques images de test, mais on ne va prendre que quatre images « .jpg »

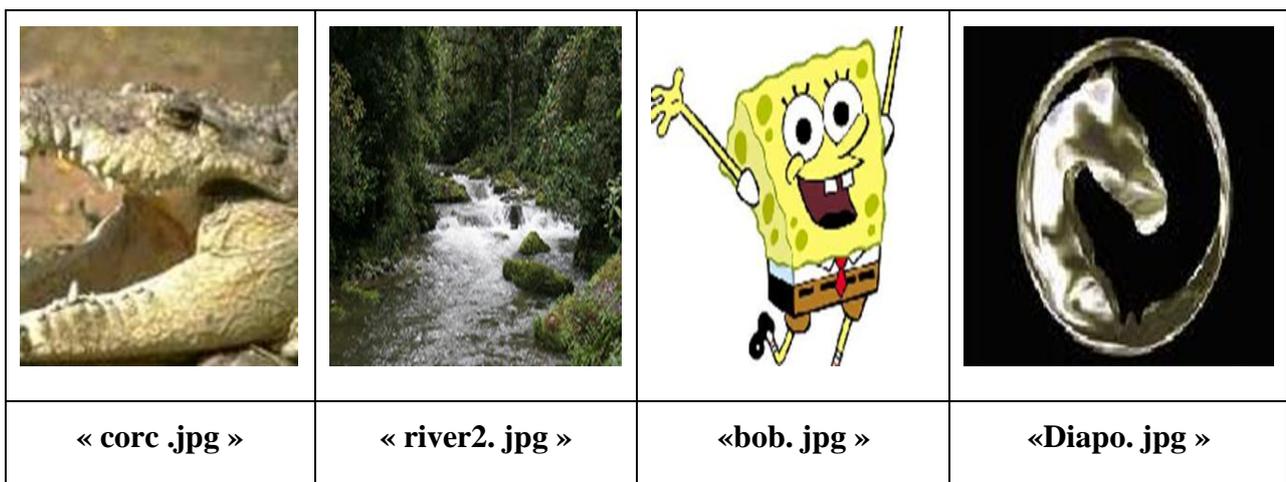


Figure 4.03: Images de test

Dans la première partie, on pratique sur l'image une transformation de Karhunen Loeve, et on obtient les trois coefficients $KL1, KL2, KL3$.

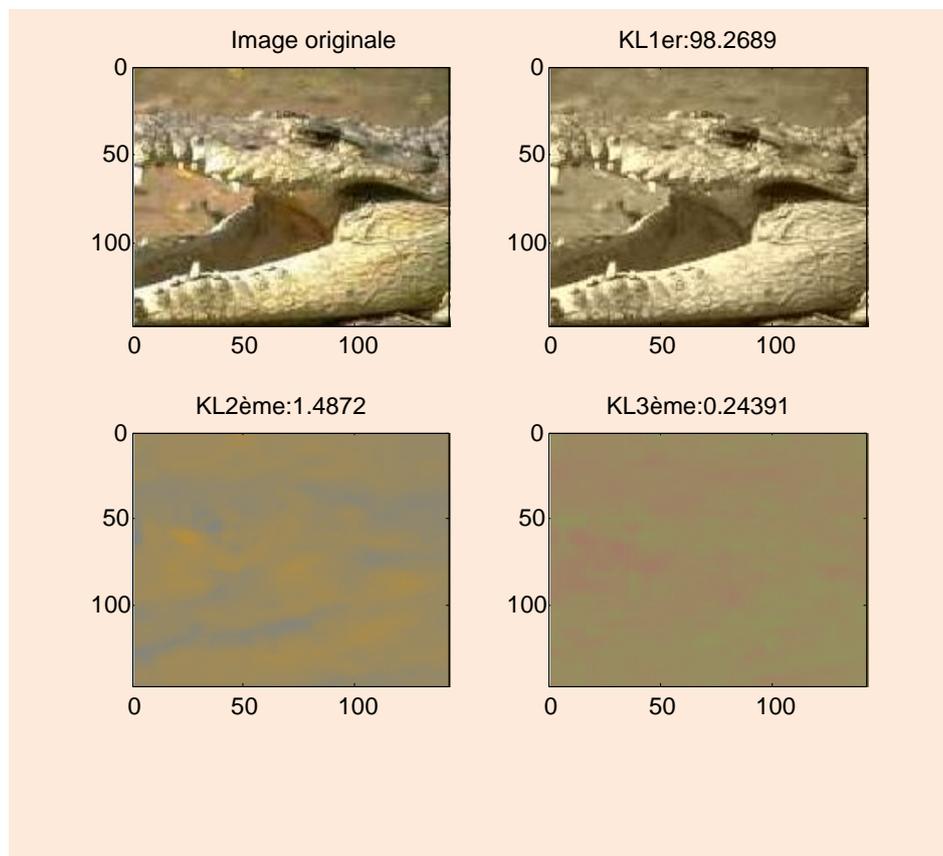


Figure 4.04 : *Transformée de l'image par KLT*

Interprétations :

On remarque que 98% des informations de l'image se concentre dans le premier coefficient $KL1$. On voit bien que la totalité de l'énergie de l'image est répartie dans les trois coefficients dont la première contient la majeure partie. Cette répartition dépend de la taille de partitionnement en bloc effectué sur l'image (512×512). Les pourcentages d'informations de l'image (d'inertie des coefficients) montrent cette variation. La réduction de la taille des blocs modifie légèrement la répartition de l'information de l'image (l'énergie).

Les deux derniers axes fournissent les 1,71 % des informations de la chrominance. C'est grâce à cette réduction qu'on affirme que la TKL est optimale pour une compression.

Le tableau 4.01 suivant récapitule les résultats du test obtenus pour des images de format « .jpg », avec les différents paramètres de compression.

Images de test	Niveau de quantification	T_c	MSE			PSNR			SSIM
« corc .jpg »	2	8	4,79	4,01	0,75	45,59	42,09	49,34	0,996
	4	8	2,02	4,12	1,04	45,05	41,97	47,92	0,994
	8	8	2,93	4,59	2,29	43,45	41,51	44,53	0,99
« river2 .jpg »	2	8	5,02	3,89	0,63	41,13	42,23	50,14	0,99
	4	8,01	5,58	4,69	1,99	46,67	41,42	42,25	0,99
	8	8,09	4,49	3,69	0,31	41,24	42,46	52,96	0,99
«bob .jpg »	2	8	17,49	15,51	0,31	35,71	36,23	53,28	0,99
	4	8	17,75	15,80	0,25	35,6	36,15	54,1	0,99
	8	8	17,96	15,89	1,13	35,59	36,12	47,61	0,99
«Diapo .jpg »	2	8,25	0,51	0,56	0,13	51,03	50,67	57,1	0,99
	4	8,25	0,52	0,51	0,39	51	51,1	52,21	0,97
	8	8,25	1,47	0,88	1,45	46,45	48,67	46,51	0,97

Tableau 4.01 : Résultats de la compression TKLH suivant les paramètres de compression

	Bloc 512*512
	MSE = 2,29
	Nombre de coefficients : KL1, KL2
	q = 8
	PSNR = 44,53
$T_c = 8$	SSIM = 0,99

Figure 4.05 : Image comprimée avec les paramètres de compression

Interprétations :

Ces valeurs sont les plus significatives dans notre résultat ; le taux de compression reste aux alentours de 8.

En particulier, la réduction de la taille diminue l'information contenue dans le premier coefficient et augmente les écarts entre les différents coefficients. Or, cet écart fait stagner le taux de compression.

Quant à la compression, elle affaiblit un peu l'efficacité de la transformation utilisée mais ce seul critère n'est pas le seul élément à considérer dans la compression.

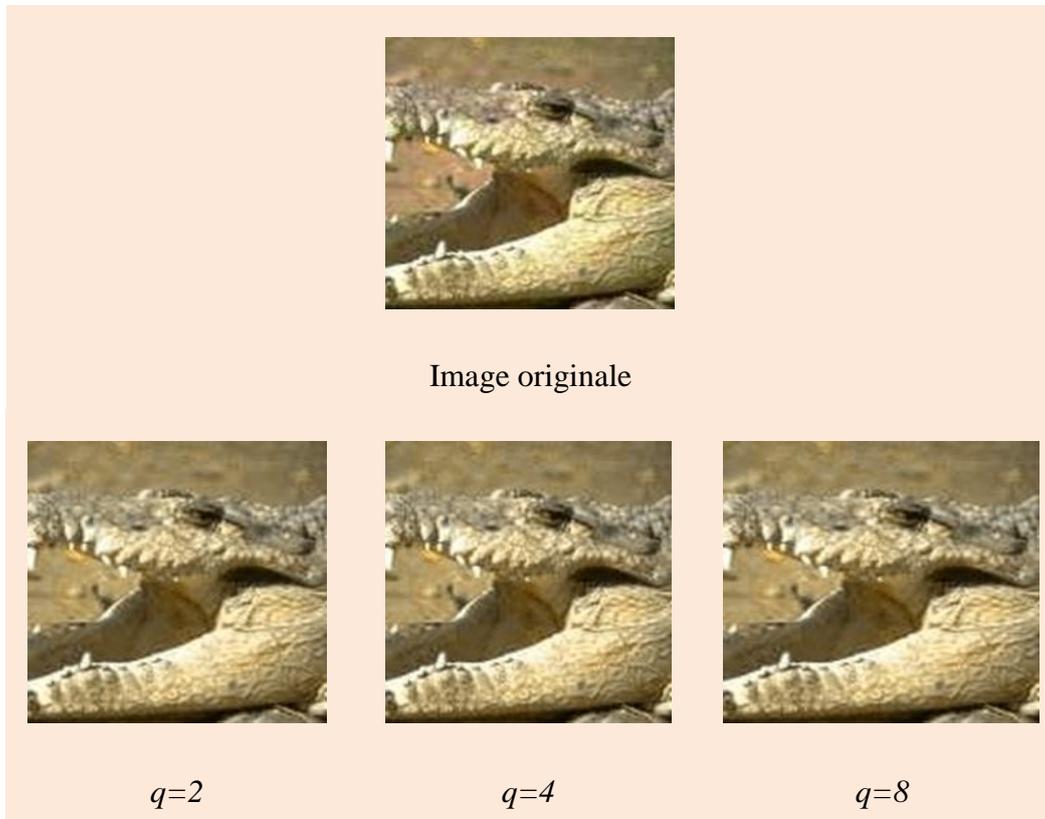


Figure 4.06 : *Reconstruction de l'image suivant les valeurs de quantification*

Pour des niveaux de quantifications variant de 4 à 8, l'image reconstruite est encore nette et similaire à l'image originale.

Ceci est rendu possible car les coefficients de KL sont regroupés dans les valeurs propres de la matrice de covariance.

Ces dernières se trouvent dans la diagonale et d'ordre décroissant. Donc, l'étape de quantification n'affecte pas trop les résultats de la compression.

– *Du point de vue pouvoir de reconstruction*

L'application de la transformation TKL, qui est optimale au sens de la décorrélation, et compactage d'énergie permettent effectivement d'améliorer, voire optimiser les techniques de compression.

On peut vérifier l'efficacité d'une transformation par sa qualité de reconstruction c'est-à-dire sa capacité plus ou moins grande à reconstruire l'image originale selon la répartition d'information de l'énergie [29].

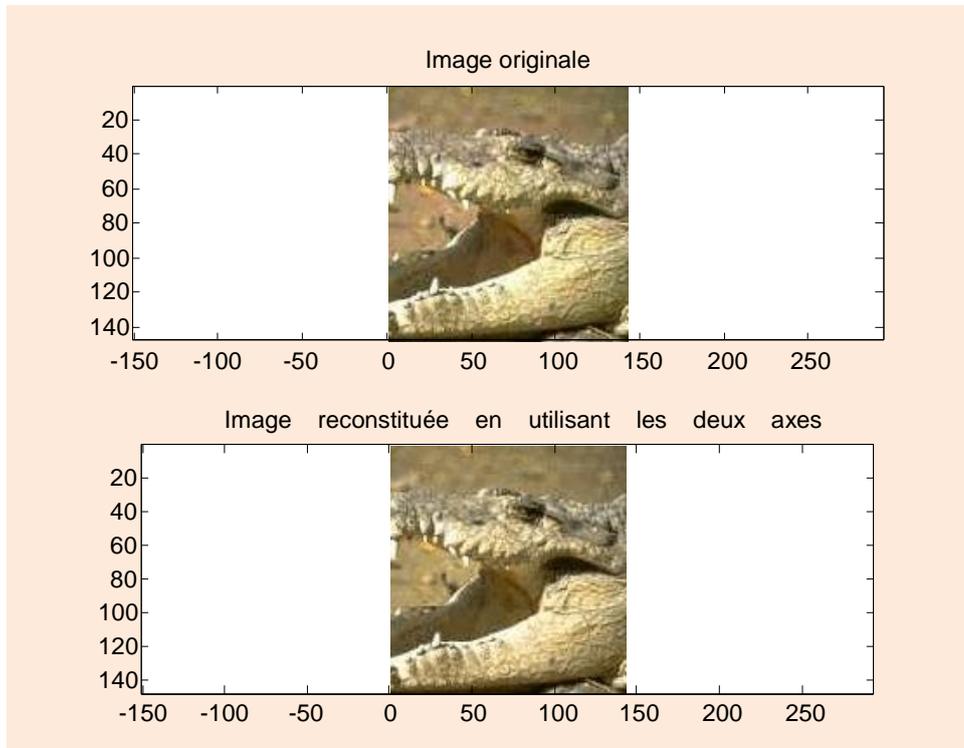


Figure 4.07 : *Reconstruction de l'image avec $KL1$, $KL2$*

Interprétations :

Il faut tenir compte du nombre de coefficients de la TKL et la taille de bloc. Pourtant, avec la TKL. Même un seul axe principal suffit pour reconstruire presque en bonne et du forme l'image originale. Ceci est confirmé par les valeurs de la SSIM et de la PSNR présenté ci-dessus.

Une manière de mesurer cette capacité est d'utiliser les métriques comme P.S.N.R et S.S.I.M qui sont des indicateurs objectifs de la dégradation de l'image reconstruite.

Pratiquement, on choisit des valeurs seuils pour les deux grandeurs, à partir desquelles on considère la dégradation acceptable.

Des plages de valeurs souvent utilisées sont 30dB à 40 dB pour le P.S.N.R et 0.8 à 0.9 pour le S.S.I.M.

4.6 Conclusion

Dans ce travail, on a pu mettre en œuvre la méthode TKL dans une chaîne de compression et l'implémenter en Matlab.

Parmi les objectifs fixés, on a pu analyser, dans le cadre de la compression par TKL, le choix du nombre de coefficients à retenir

Evaluer les performances d'une chaîne de compression par TKL intégrant un quantificateur optimal et d'un codeur entropique qui consistent à tester les performances de la méthode, nous avons pu vérifier et constater que :

L'utilisation d'une transformation optimale au sens de la décorrélation et compactage d'énergie permettent effectivement d'améliorer, voire optimiser les techniques de compression.

Cependant, une transformation optimale telle la TKL ne possède pas un algorithme rapide comme celui de TCD. Ce qui ne rend pas pratique cette méthode par rapport à l'utilisation de TCD pour Jpeg.

Du point de vue performance, en faisant un bon choix des paramètres de compression tels que taille de bloc, nombre de coefficients utilisées, nombre de niveau de quantification, la méthode TKL peut réaliser un taux de compression marquante avec une qualité d'image reconstruite similaire.

CONCLUSION GENERALE

Même si la compression d'images n'est pas une récente apparition dans le domaine du traitement d'images, on peut dire que de nombreux ouvrages ou publications font encore l'objet de recherches.

On peut aisément se rendre compte de la pleine expansion de ce domaine. Et ce aussi bien au niveau des techniques de codages qui sont de plus en plus variées, mais aussi au niveau de l'efficacité des algorithmes utilisés en termes de complexité. La mise en œuvre de la compression d'images utilisant la TKL permet effectivement une réduction de la dimensionnalité, d'améliorer les techniques de compression.

Dans ce mémoire, nous avons pu mettre en œuvre la méthode TKL dans une chaîne de compression et l'implémenter en Matlab.

Du point de vue performance, en faisant un bon choix des paramètres de compression tels que le nombre de coefficients à prendre, ou le nombre de niveau de quantification, ou la taille du bloc d'image à subdiviser.

Cette méthode peut réaliser suivant une image donnée, un taux de compression notable avec une qualité d'image reconstruite assez similaire.

Cependant, cette transformation ne possède pas un algorithme rapide. L'algorithme de la transformation de Karhunen Loeve nécessite un temps de calcul important, ce qui pénalise la vitesse de la méthode. La transformation de Karhunen Loeve n'a pas la prétention d'être un outil universel. On a rencontré des images pour lesquelles la compression n'a pas donné le résultat attendu.

Pour compresser correctement les images, il faut que l'inertie expliquée sur les deux premiers axes soit supérieurs à 98% car le nuage des couleurs se présente sous une forme aplatie, ce qui est favorable à la compression des données.

Même si on n'a pas utilisé des protocoles de test rigoureux, les résultats obtenus restent concluants en tenant compte des méthodologies adoptées. L'étude effectuée dans le cadre de ce mémoire nous permet d'envisager d'appliquer la méthode TKL dans le domaine de l'indexation, ou à une méthode de compression vidéo. Dans ce dernier cas, la TKL devrait être plus efficace, pour réduire au maximum les redondances temporelles entre les différents frames de la séquence vidéo.

ANNEXE 1 NORME JPEG

A1.1 Définition de la Normes JPEG

Le format JPEG, très couramment utilisé pour le codage des images bitmap et des photos, est un format de compression très efficace.

La perte de qualité d'image occasionnée par l'algorithme de compression peut être maîtrisée car le taux de compression des fichiers .JPEG est réglable [14].

A1. 2 Présentation de la Norme JPEG

Le sigle JPEG, aussi appelé JPG, est l'acronyme de Joint Photographic Experts Group.

Ce sigle, qui faisait à l'origine référence à un comité d'une trentaine de professionnels de l'image, a donné son nom à une norme ouverte de compression d'images numériques.

Aujourd'hui, le sigle JPEG est également employé pour définir un format de données, ainsi que le format de fichier le plus couramment utilisé pour contenir ces données.

La norme de compression JPEG, aussi connue comme la norme ISO/CEI 10918-1 ou UIT-T Recommandation T.81, concerne uniquement l'algorithme et le format de décodage des images.

Elle a été spécifiée par le comité du même nom en 1991 et officiellement adoptée sous sa forme définitive en 1992, à l'issue de plus de 10 ans de travaux et d'essais de compression d'images.

Cette norme ne spécifie pas la méthode de stockage des informations relatives à une image (dimensions, auteur, etc.). Cela correspond au rôle du format de fichier JPEG File Interchange Format généralement abrégé JFIF. L'extension de fichier .JPEG ou .JPG correspond en réalité à un fichier au format JFIF.

Il s'agit du format de fichier le plus couramment employé pour les images enregistrées avec la compression JPEG. Ce format ne gère pas la transparence et ne permet pas de créer des animations.

A1.3 Caractéristiques de la norme JPEG

Le format JPEG permet de compresser les images bitmap très détaillées, en utilisant un algorithme de compression destructif.

Cet algorithme entraîne en effet une dégradation de la qualité d'image pour offrir des taux de compression élevés.

On distingue deux types de processus de compression :

- la compression avec pertes ou compression irréversible autorise des taux de compression importants (elle permet d'obtenir des fichiers jusqu'à 20 fois moins volumineux)
- la compression sans pertes ou compression réversible permet de revenir aux valeurs originales de l'image puisqu'il n'y a pas de pertes d'information.

En contrepartie, le taux de compression est moindre. Ce deuxième type de compression fait l'objet d'une norme spécifique appelée JPEG-LS.

Les fichiers .Jpeg utilisent le standard informatique pour le stockage des métadonnées (titre, auteur, agence, copyright, etc.) défini par l'IPTC pour les images de presse : l'IPTC Core.

A1.4 Avantages et inconvénients du format JPEG

Le format JFIF, plus connu sous le nom de format JPEG, est complémentaire des formats GIF et PNG pour la publication d'images sur le Web.

Il sauvegarde plus d'informations couleur que le format GIF et permet de comprimer des photographies ou des images lourdes.

L'algorithme de compression provoque une perte d'information et peut donc entraîner une perte de qualité visible quand on utilise un taux de compression élevé ou sur certains types d'images.

Ainsi, le format .jpeg convient bien aux photos mais pas aux images comportant peu de couleurs ou aux figures géométriques (pour lesquelles le format GIF est plus adapté).

Le principal avantage de ce format est le taux de compression réglable qui permet à l'utilisateur de trouver un compromis entre le taux de compression et la qualité de l'image

A1. 5 Normes JPEG 2000

A1.5.1 Etapes de transformation

La norme JPEG2000 est l'évolution de la norme JPEG. Son schéma synoptique est illustré par la figure A1.01.

Le montage en amont constitue la chaîne de compression et celui en aval représente celle du décodage.

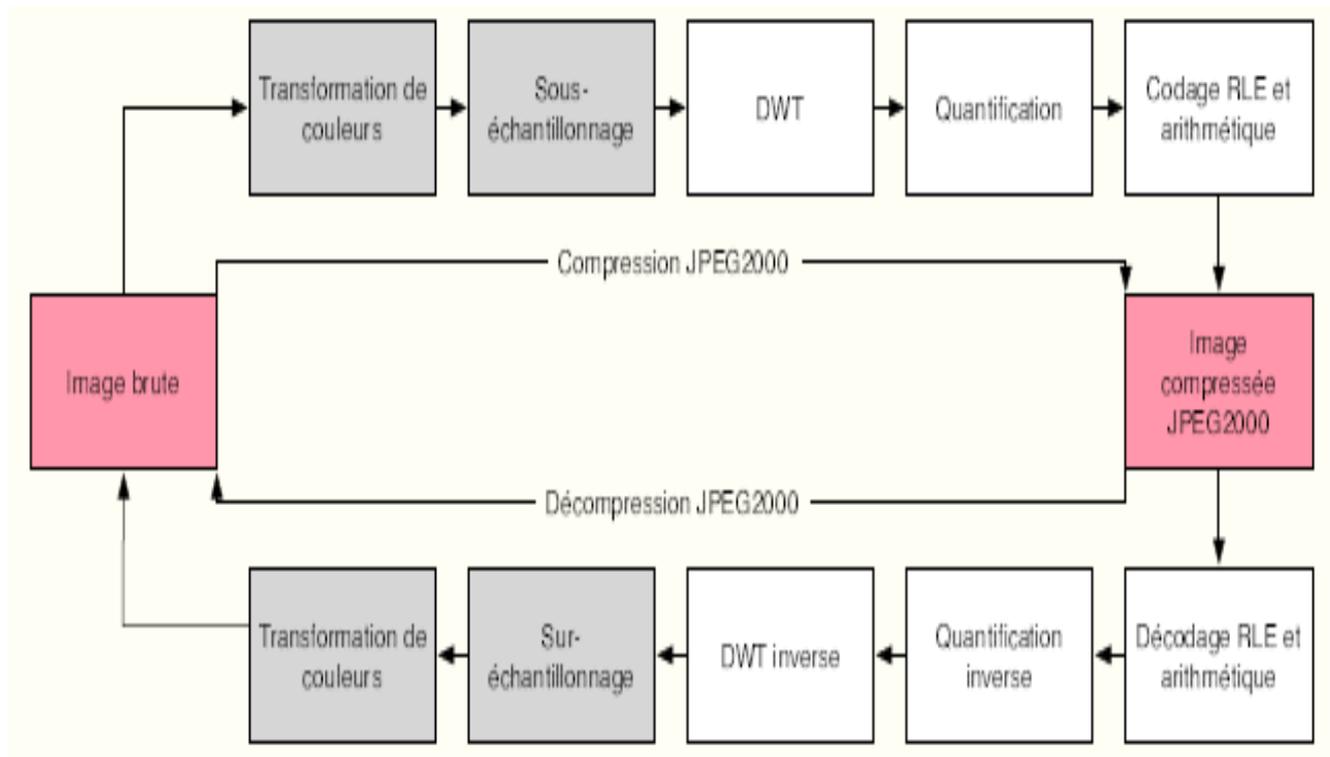


Figure A1.01 : *Etapes de transformation dans la norme JPEG 2000*

A1.5.2 Transformation des plans de couleur et sous-échantillonnage

La transformation des plans de couleur et le sous-échantillonnage de ceux-ci sont identiques à ceux définis précédemment dans la compression *JPEG*.

A1.5.3 DWT – Discrete Wavelet Transform

DWT signifie littéralement Transformée en Ondelettes Discrète.

Elle est à la base de la transformation utilisée dans la norme JPEG2000.

Elle est issue d'une analyse temps-fréquence d'un signal. Elle est appliquée sur l'intégralité de l'image et non plus sur chaque bloc comme dans la norme JPEG.

Principe

Le principe d'incertitude est à la base de la transformée en ondelettes. Il est issu des résultats de la transformée de Fourier.

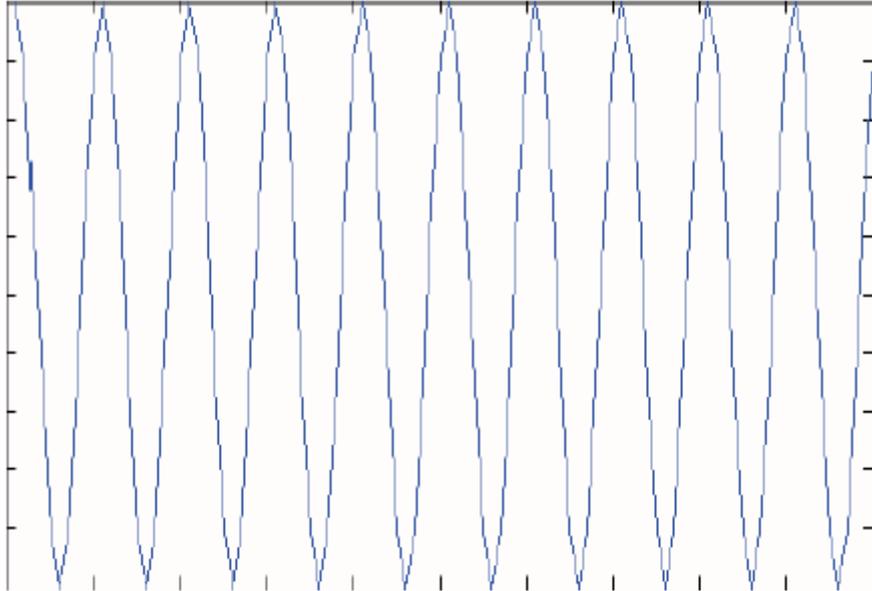


Figure A1.02 : *Fonction sinusoidale*

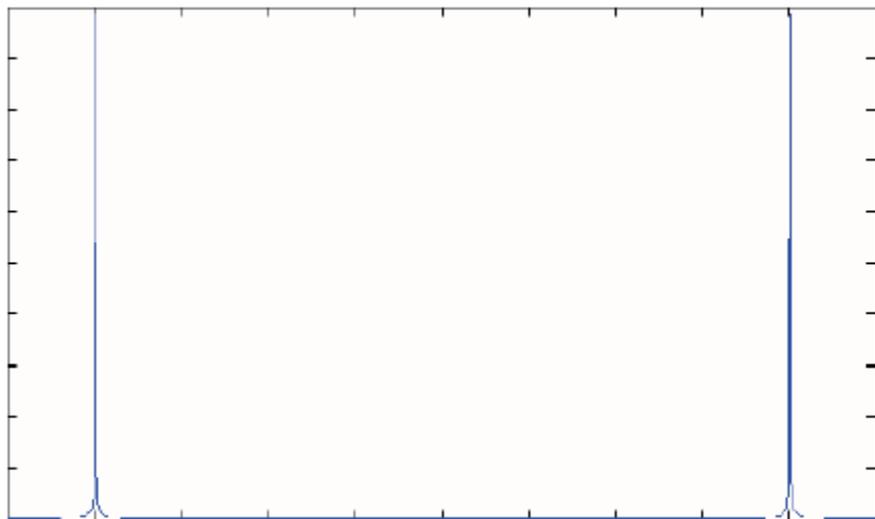


Figure A1.03 : *Représentation fréquentielle d'une fonction sinusoidale*

En effet, dans la transformée de Fourier, une fréquence pure (connaissance fréquentielle locale) détermine une fonction temporelle sinusoidale (connaissance temporelle globale).

Inversement, un signal temporel défini par un pic de Dirac (connaissance temporelle locale) détermine une fonction fréquentielle (connaissance fréquentielle globale).

On note ainsi que la connaissance précise d'une caractéristique d'un signal dans un domaine (par exemple, temporel) renseigne sur sa connaissance globale dans l'autre domaine (fréquentiel).

Ce principe a amené à s'intéresser aux propriétés locales d'un signal `a travers l'analyse temps-fréquence ou multi-résolution.

On va ainsi s'intéresser aux propriétés fréquentielles locales d'un signal à chaque instant.

A1.5.3.1 Ondelettes

Les ondelettes représentent une famille de fonctions particulières qui sont `a la base de la DWT.

Une ondelette $\psi(t)$ possède deux propriétés :

- L'aire sous sa courbe est nulle
- Son énergie est finie

A1.5.3.2 Apport de la DCT

Après transformation par DWT d'une image, on obtient une autre image de même taille ou les basses fréquences spatiales de l'image sont situées dans le coin supérieur gauche (moyenne de l'image originale).

Les fréquences spatiales augmentent `a mesure que l'on prend des sous-imagettes qui se rapprochent du coin inférieur droit.

On obtient alors une représentation fréquentielle proche de celle du JPEG ou les coefficients apportant le maximum d'informations se trouvent dans les sous-imagettes proches du coin supérieur gauche. Les autres coefficients de détail sont négligeables.

A1.5.4 Quantification

Tout comme avec la DCT, après la DWT le signal bidimensionnel représentant l'image se présente sous une nouvelle forme.

Cependant, la transformation est conservatrice: la DWT inverse permet de retrouver l'image originale sans aucune altération.

Le principe de quantification reste le même que pour le JPEG. La quantification n'est pas faite sur les blocs de 8×8 pixels mais sur chacune des sous-imagettes.

Le pas de quantification diminue `a mesure que l'approximation est de plus en plus grossière pour avoir une plus grande précision sur les valeurs des basses fréquences spatiales.

A1.5.5 Codage RLE et Huffman

La transformée en ondelettes possède des propriétés identiques à celle de la transformée par *DCT* dans la mesure où la répartition spatiale des hautes fréquences et des basses fréquences dans l'image résultat est identique. Les techniques d'encodage d'une image résultant d'une *DWT* sont donc identiques à celles définies pour le *JPEG*.

ANNEXE2 CODAGE D'IMAGE

A2.1 Type de codage

En général il existe 02 grandes catégories de codage:

- Le codage réversible
- Le codage irréversible

A2.1.1 Le codage réversible

Le codage est dit réversible s'il y a interdiction d'altérer les données. Le codage réversible ou sans perte implique que l'image obtenue après décodage de l'image comprimé reste le même que l'image originale $I = D(C(I))$.

Cette méthode est généralement utilisée lorsque les données ne doivent subir aucune altération, à cause de leur nature ou bien du contexte de codage [15].

A2.1.2 Le codage irréversible

Les méthodes irréversibles sont également basées sur la redondance présente dans l'image. En même temps, elle exploite le fait que l'œil humain n'est pas sensible à certaine dégradation. Par exemple, le nombre de niveau de gris que l'œil peut distinguer est largement inférieur à 256. Quant au couleur, l'œil est largement incapable d'en distinguer autant de couleur (une image couleur de 24bits contient 256^3 couleurs possible). La quantification est la base pour les méthodes irréversibles [14].

A2.2 Méthodes de codage

Il y a plusieurs méthodes de codage de données que l'on peut regroupée en deux :

- Les codages entropiques :
 - o Méthodes Spatiales
 - o Codage Statistique
- Les codages par transformées :
 - o Transformations Orthogonales

- Méthodes hiérarchiques
- Décomposition par bancs de filtres

A2.3 Codage entropique

Soit X une source aléatoire prenant ses valeurs dans un alphabet finie de K symboles $A = \{x_k\}_{1 \leq k \leq K}$. Le but est de minimisé le nombre de bit moyen pour stocké les valeurs de X .

On considère d'abord des codes qui à chaque symbole x_k , associe un mot binaire w_k de longueur l_k .

Une suite de valeur produite par la source X est codée par concaténation des mots binaires correspondant.

Tous les symboles x_k peuvent être codés avec des mots binaires de la même taille $l_k = \log_2(K)$ arrondie à l'entier supérieure.

Cependant le nombre moyen de bit par symbole peut être réduite en utilisant des mots plus courts pour les symboles les plus fréquents.

La probabilité d'apparition d'un symbole x_k est :

$$p_k = P_r[X = x_k] \quad (\text{A2.01})$$

Le nombre moyen de bit nécessaire au codage de chaque symbole émis par la source X est donné par :

$$R_X = \sum_k l_k p_k \quad (\text{A2.02})$$

Notre but est d'optimiser les mots w_k du code afin de minimiser R_k .

A2.3.1 Théorème de Shannon

Soit X une source dont les symboles $\{x_k\}$ avec $1 \leq k \leq K$ ont des probabilités d'apparition $\{p_k\}$ avec $1 \leq k \leq K$.

Le nombre moyen de bit d'un code préfixe vérifie $R_X \leq H_X = -\sum p_k \log_2 p_k$.

De plus, il existe un code préfixe tel que $R_X \leq H_X + 1$ et la somme H_X s'appelle entropie de X .

Un code préfixe vérifie donc l'équation $H_X \leq R_X \leq H_X + 1$.

L'entropie H_X mesure l'incertitude portant sur le résultat de la variable aléatoire X , et sa valeur est comprise entre $0 \leq H_X \leq \log_2(K)$. La valeur maximale $\log_2(K)$ correspond à une distribution uniforme $p_k = \frac{1}{K}$.

La valeur minimale « 0 » correspond à un symbole dont la probabilité d'apparition est « 1 ».

Le codage de Huffman est un algorithme de codage de type entropique basé sur les fréquences d'apparition des pixels de l'image original. Introduit en 1952 par David A. Huffman un étudiant du MIT (Massachusetts Institute of Technology [16]).

Cette technique est largement utilisée car elle est très efficace, observe selon le type de données des taux de compression allant de 20% à 90% mais plus généralement entre 30% et 60% [14].

L'algorithme de Huffman produit des codes optimaux, dans la mesure où il produit des codes de longueur moyenne minimale. Et bien qu'il puisse exister un certain nombre de codes équivalents ; aucune procédure ne peut produire des codes plus courts.

En moyenne, sous les mêmes conditions, à savoir que les codes individuels sont de longueur variable mais entière [15].

A2.3.2 Codages statistiques

Le codage statistique tient compte de la fréquence d'apparition d'un symbole dans l'ensemble de l'image.

Ceci, afin de coder sur un « nombre de bits réduit » les symboles les plus probables.

Ces symboles se trouvent sur des structures binaires plus longues les symboles les moins fréquents. Ce type de codage est limité par l'entropie H de l'image. On peut trouver comme type de codage statistique : l'algorithme de codage de Huffman et l'algorithme Lempel Ziv Welch.

A2.3.3 Codage par la méthode spatiale

La méthode spatiale consiste à réduire l'information redondante en tenant compte de la corrélation spatiale des éléments à coder. On va voir par exemple le codage par plage ou RLE.

A2.3.3.1 Codage de type RLE

RLE ou Run Length encoding est un algorithme de compression de données qui est utilisé par la plupart des formats de fichiers bitmaps tels que TIFF et BMP.

Il a été créé pour compresser n'importe quel type de données sans tenir compte de l'information qu'elle contient. Toutefois, le contenu des données va affecter le taux de compression qu'il pourra atteindre. Bien que la plupart des algorithmes RLE ne puissent pas atteindre les forts taux de compression, d'autres méthodes de compression plus avancées, RLE est à la fois simple à implémenter et rapide d'exécution ; ce qui fait de lui une bonne alternative, entre utiliser un algorithme de compression plus complexe ou laisser l'image sans compression [16].

a) Principe

RLE est l'une des méthodes les plus anciennes, des plus simples et la plus utilisée. Cette méthode consiste à identifier et supprimer des redondances d'informations en les codant sous une forme plus compacte. RLE s'emploie à réduire la taille physique d'une répétition de chaîne de caractère. Cette chaîne répétée est appelée un passage (run) et est typiquement codée avec 2 octets.

Le premier byte représente le nombre de caractères dans le passage et le deuxième est appelé le compteur de passage (run count).

b) Algorithmes

- Encodage

```
Prendre une image
Initialiser le compteurOccurrence
Initialiser à nulle le pixelRéférence
Initialiser pixelCourant au premier pixel de la lecture
Commencer la lecture
Tant que : non fin lecture
    Si (pixel courant = pixel de référence)
        compteurOccurrence + 1
    Sinon
        pixelCodé = (pixelRéférence,compteurOccurrence)
        pixelRéférence = pixelCourant
        compteurOccurrence = 0
    FinSi
    pixelCourant devient le prochain dans la lecture
Fin Tant Que
```

- Décodage

```
Prendre l'image codée
Tant que : non fin lecture
    compteurOccurrence = pixelCodé(compteurOccurrence)
    Pour i de 1 à compteurOccurrence pas de 1
        pixelCourant = pixelCodé(pixelCourant)
    Fin Pour
    pixelCodé courant devient le prochain dans la lecture
Fin Tant Que
```

c) Avantages et inconvénients

Le principal avantage de l'encodage RLE est son immense simplicité. Son algorithme tient sur quelques lignes à peine autant au niveau du codage que du décodage.

Il peut de plus être utilisé sur n'importe quel type de fichier ou après d'autres algorithmes de compression plus évolués pour gagner quelques octets de plus.

En revanche, l'encodage RLE possède un désavantage de taille. L'image doit contenir plusieurs pixels répétés pour être d'une quelconque utilité.

C'est pourquoi, on le voit très souvent précédé d'une transformation (exemple DCT) qui reclasse l'image afin d'augmenter les chances d'avoir plus de caractères ou bits répétés.

A2.4 Codage par transformation orthogonale

Une transformation orthogonale réalise une rotation de l'espace de représentation (par exemple spatiales ou fréquentielles).

L'image subit une transformation permettant de mettre en évidence ses structures, de le modéliser et de mettre en évidence les redondances présentes dans l'image.

Les coefficients décorrélés de l'image transformée détiennent alors de façon quasi unique et indépendante les caractéristiques de l'image.

Si ces informations sont classées par ordre d'importance, une compression des données peut être obtenue, d'une part par l'élimination des coefficients nuls (voire de faible amplitude) et d'autre part, par une quantification plus ou moins grossière des coefficients sélectionnés pour la transmission. On peut trouver: la transformée de Fourier discrète et la transformée en cosinus discrète.

A2.5 Codage par la méthode hiérarchique

Cette classe de méthode de codage permet la reconstruction progressive, ou l'accès aux images de qualité et de résolution différentes.

L'exemple souvent cité est celui de la recherche d'images compressées dans une base de données.

Le codage hiérarchique est aussi utile lorsque l'image compressée est utilisée par différents équipements de visualisation ou de transmission :

- écran de télévision haute définition (HDTV) avec incrustation d'écran de moindre qualité,
- image sortie sur une imprimante de très haute résolution,
- ou transmise sur une ligne téléphonique à basse résolution. On trouve le codage fractal dans la méthode hiérarchique.

ANNEXE 3 LOGICIEL MatLab

A3.1 Présentation du logiciel MatLab

MatLab est un langage très performant pour le calcul technique. Il intègre le calcul, la visualisation, et la programmation dans un environnement facile à utiliser dans lequel les problèmes et les solutions sont exprimés en notations mathématiques familières.

Les utilisations typiques incluent:

- les calculs mathématiques
- le développement d'algorithmes
- l'acquisition de données
- la modélisation, la simulation, et le prototypage
- l'analyse de données, l'exploration, et la visualisation
- les graphes scientifiques
- le développement d'applications, y compris la construction d'interface utilisateurs graphiques (GUI)

MatLab est un système interactif dont l'élément d'informations de base est une matrice qui n'exige pas le dimensionnement.

Ceci permet de résoudre beaucoup de problèmes de calcul technique, particulièrement ceux avec des formulations de matrice et de vecteur, d'autre part, il permet aussi d'écrire un programme dans un langage non-interactive telle que C ou Fortran.

Le sigle MatLab veut dire « *Matrix Laboratory* ». MatLab a évolué pendant des années avec l'entrée de beaucoup d'utilisateurs.

Dans les universités, c'est l'outil d'instruction standard pour des cours d'introduction et de perfectionnement des mathématiques, de la technologie, et de la science.

Dans l'industrie, MatLab est l'outil du choix pour la recherche de haut-productivité, développement, et analyse.

MatLab comporte une famille des solutions spécifiques à l'application ajoutées appelées les boîtes à outils qui sont importants pour la plupart des utilisateurs de MatLab.

Ces boîtes à outils permettent d'apprendre et d'appliquer des technologies spécialisées.

Les boîtes à outils sont des collections complètes de la fonction MatLab (les fichiers .m) qui prolongent l'environnement de MatLab pour résoudre des problèmes particuliers.

Les secteurs dans lesquels les boîtes à outils sont disponibles incluent le traitement des signaux, commande des systèmes, réseaux des neurones, logique floue, ondelettes, simulation, et beaucoup d'autres.

BIBLIOGRAPHIE

- [1] M. A. Rakotomalala, « *Cours outils et méthodes de traitement d'image* », 4^{ème} Année ; Département télécommunication, Université d'Antananarivo, année scolaire 2011-2012.
- [2] F. Tupin, « *Cours OASIS* », Université EPFL, Département Signal et Image, Année 2007-2008.
- [3] M. Pastoureau, « *Histoire d'une couleur* », éditions du seuil, 2002.
- [4] M. Kunt, G Granlund, M. Kocher, « *Traitement Numérique des Images* », Presses polytechniques et universitaires romandes, Lausanne 1993.
- [5] P. Kowaliski, « *Vision et mesure de la couleur* », Physique fondamentale et appliquée. Masson, 2ème édition, 1990.
- [6] M. Hadallah, « *Codage des images fixes par méthodes des hybride basée sur la QV et les approximations* », 2005.
- [7] I. Bloch, Y. Gousseau, H. Maitre, D. Matignon, B. Pesquet-Popescu, F. Schmitt, M. Sigelle, F. Tupin, « *Le traitement des images* », tome1, Telecom-Paris, 2005.
- [8] A. Zenoni, « *Mesure de la qualité des images agrandies par décomposition en ondelettes* », DEA INP, Grenoble, 2003.
- [9] J.P Guillois, « *Compression des données, compression des images* », techniques de l'ingénieur, 2000.
- [10] P. Lambert, « *Etudes méthodologiques du filtrage et de la segmentation d'images multi-composantes* », H.D.R université de Savoie, 2002.
- [11] A. Gersho, R. M Gray, « *Vector quantization and signal compression* », Boston, Kluwer academic publishers, 1992.
- [12] T. Berger, « *Minimum entropy quantizers and permutation codes* », IEEE Trans, On Information Theory, vol 28, 1982.
- [13] S. Peterson, « *Algorithmic information theory* », Mathematics of Digital Information Processing, Springer 2006.
- [14] O. Rioul, P. Duhamel, « *Information et codage* », Module ES 204, ENSTA, 2001.
- [15] L. Seoud, « *Introduction aux technologies multimedia* », Ecole polytechnique de Montréal, Département de génie informatique, 2009.
- [16] S. Thon, « *Codage et compression des images 2* », Institut Universitaire de Technologie, Aix Marseille, 2013-2014.

- [17] R.C. Gonzales, R. E. Woods, « *Digital Image Processing* », 3rd Edition, University, of Tennessee, 2004.
- [18] B. Budin, « *Métriques perceptuelles pour la compression d'images* », Master en sciences, Université de Laval Québec, 2007.
- [19] M. Kramm, « *Compression of Image Clusters using Karhunen Loeve transformations* », Munchen Germany, Institute for computer science, Boltzmann, 2006.
- [20] M. H. Ramafiarisona, « *Modélisation d'un système de recherche d'images par le contenu avec l'analyse factorielle de données dans une base hétérogène* », Thèse de doctorat en télécommunication, Université d'Antananarivo, 2013.
- [21] A. B. Buisson, V. Lattuati, D. Lemoine, « *Présegmentation d'images couleur par la transformée de Karhunen Loeve* », Laboratoire d'Automatique des Arts et de Métiers CNAM, quatorzième colloque Gretsi, Juan les pins, 2003.
- [22] A. K. Jain, « *A fast Karhunen Loeve transform for a class of random processes* », Digital Image Processing & Analysis- Vol 1- IEEE Computer Society, 2003.
- [23] I.A.L. Ocadiz, « *Analyse en composantes principales d'une image couleur* », Thèse de l'institut Polytechnique de Grenoble, 2004.
- [24] H.H. Ralaivao, « *Études des performances de l'A.C.P dans une compression d'images couleurs fixes* », mémoire d'études approfondies en télécommunication, Département Télécommunication, Université d'Antananarivo, 2012.
- [25] Clarke, « *Digital compression of still images and video* », Academic press, San Diego, 1995.
- [26] Rao, « *Handbook of the transform & data compression* », 2009.
- [27] AlBovik, « *Handbook of Image and video processing* », Academic press: 2000.
- [28] K. Yeung, « *Principal Component Analysis for clustering gene expression data* », University of Washington, Seattle, technical report UWCSE, 2000.
- [29] K. S. Thyagarajan, « *Still Image and Video Compression with Matlab* », John Willey & Sons, 2001.
- [30] V. Ricordel, « *Compression d'image format JPEG* », <http://serecom.univ-tln.fr/~ricordel>.
- [31] MathWorks, <http://www.mathworks.com>.