



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Amar Thelidji- Laghouat

**FACULTE: DE TECHNOLOGIE
DEPARTEMENT D'ELECTRONIQUE**

MEMOIRE DE MASTER

Réalisé par : HAMMOUDI MABROUK

DOMAINE : Science et Technologie

FILIERE : Electronique

OPTION : Electronique des systèmes embarqués

Thème

Compression des images par LBG

Jury de soutenance :

Nom et Prénom	Grade	Qualité
DEBAGH MAZEN	MCA	Encadrant
SILAA MOHAMMED YOUSRI	MAA	Président
BIRANE ABDELKADER	MAA	Examineur

Promotion : 2021/2022

Remerciements

Tout d'abord, je voudrais remercier le Dieu
Tout-Puissant et Miséricordieux pour la
patience qu'Il m'a accordée tout au long de
mon travail.

Je remercie sincèrement le professeur
Merah Lahcen et **Debagh Mazen**

Pour les conseils qu'ils m'ont toujours
prodigués, pour les connaissances qu'ils me
transmettent, leur soutien attentif et leur
grande confiance en moi.

Ma plus profonde gratitude à tous les
professeurs du département d'électronique
qui m'ont encouragé à donner le meilleur de
moi-même et à m'assurer d'avoir la meilleure
formation possible.

Un remerciement spécial aux étudiants de
MASTER2 d'électronique système
embarqué.

Dédicace

Je dédie ce travail à :

**A ceux qui sont les plus chers au monde,
ma mère et mon père, à qui je n'arriverai
jamais à exprimer ma gratitude, pour leur
amour et leur soutien tout au long de mes
études.**

**Un cadeau spécial: Ma grand-mère et mon
oncle Rahmat Allah sur eux**

Famille et amis: De près ou de loin

Sommaire

Remerciements	I
Dédicaces	II
Liste des figures	III
Liste des tableaux	V
Liste D'acronymes	VI
Introduction générale.....	01
Chapitre I : Notions traitement d'image	
<u>I.1. Introduction</u>	3
<u>I.2. Définition d'une image</u>	3
<u>I.3. Notions fondamentales en traitement</u>	4
<u>I.3.1 Pixel</u>	4
<u>I.3.2 Voisinage d'un pixel</u>	4
<u>I.3.4 Image numérique</u>	5
<u>I.3.5 Résolution d'une image</u>	6
<u>I.3.6 Poids d'une image</u>	6
<u>I.3.7 Image en niveaux de gris</u>	6
<u>I.3.8 Adaptation à luminance et sa distinction</u>	7
<u>I.3.9 Image simple</u>	10
<u>I.3.10 L'image en couleurs</u>	1Error! Bookmark not defined.
<u>I.3.11 Bruit dans l'image</u>	1Error! Bookmark not defined.
<u>I.4. Nature des images</u>	Error! Bookmark not defined.2
<u>I.5. Domaines d'applications du traitement d'images</u> Error! Bookmark not defined.2	
<u>I.6. Etape de l'analyse d'une image</u>	Error! Bookmark not defined.3
<u>I.7. Conclusion</u>	Error! Bookmark not defined.3

Contents

<u>II.1 Introduction</u>	16
<u>II.2 Définition de la compression</u>	16
<u>II.3 Principe général de la compression d'image</u>	17
<u>II.4 Les deux principes de base méthodes de compression</u>	17
<u>II.5 Echantillons et quantification</u>	18
<u>II.5.1 Echantillons et quantification régulier</u>	18
<u>II.5.2 Nombre d'une unité image</u>	20
<u>II.6. Quelques relations de base entre les unités picturales</u>	21
<u>II.6.1 Faux voisins d'unité d'image</u>	21
<u>II.6.2 Conductivité</u>	22

Sommaire

<u>II.6.3 Opérations arithmétiques</u>	23
<u>II.7. La quantification vectorielle et codebook</u>	25
<u>II.8 CDEBOOK GENERATION ALGORITHMS</u>	29
<u>II.8.1. Linde, Buzo and Gray (LBG) Algorithm</u>	29
<u>II.8.2 Linde-Buzo-Gray algorithm Convert RGB image</u>	30
<u>II.8.3 Phase d'extraction</u>	31
<u>II.9 Équations et éléments d'algorithme (LBG)</u>	31
<u>II.9.1 Équations</u>	31
<u>II.9.2 Éléments d'algorithme avec (LBG)</u>	33
<u>II.10 Les avantages et Les inconvénients</u>	34
<u>II.11 CONCLUSIONS</u>	35

Chapitre III: Compression des images par l'Algorithmes

LBG Simulation Et Résultats

<u>III.1. Introduction</u>	37
<u>III.2. introduction au programme MATLAB</u>	37
<u>III.3. Organigramme de l'algorithme (LBG)</u>	38
<u>III.4.Résultats des simulations</u>	44
<u>III.4.1 code</u>	44
<u>III.4.2 Présentation de l'interface graphique</u>	45
<u>III.5 Testes et Résultats de la compression</u>	50
<u>III.6 Conclusion</u>	58
Conclusion Générale.....	60
Bibliographie	61
Résumé	65

Liste des figures

Chapitre I : Notions traitement d'image

Figure I.1 : Un point de l'image de coordonnées	3
Figure I.2: Représentation de lettre A sous la forme d'un groupe de pixels	4
Figure I.3: Voisinage d'un pixel	5
Figure I.4: matrice	5
Figure I.5 : Valeurs des niveaux de gris et teintes de gris correspondantes	7
Figure I.6: Exemple d'une image en niveaux de gris	7
Figure I.7: Relation entre l'intensité de l'éclairage entrant dans l'œil et la luminance	8
Figure I.8: Valeurs des niveaux de gris et teintes de gris correspondante	8
Figure I.9: montre la nette différence de niveaux de gris	9
Figure I.10: Une explication du processus de contraste	9
Figure I.11: Image numérisée avec 16 millions de couleurs	11
Figure I.12: Histogramme d'image	12

Chapitre II : Techniques compression d'images par LBG

Figure II.1: Schéma synoptique de compression	17
Figure II.2: Une explication du processus de contraste	19
Figure II.3: (A) Unités factices régulièrement espacées. (B) Unités picturales irrégulières	20

Liste des figures

Figure II.4: Unités d'image et ses voisins	21
Figure II.5.: : Une coupe transversale des unités d'image de la matrice d'image unité d'image Nominal $m(x, y)$ Dans une image, les quatre voisins par position de coordonnées	21
Figure II.6: Unité d'image $p(x, y)$ dans une image et les pays voisins répartis en fonction de l'emplacement des coordonnée.....	22
Figure II.7: (A) Un groupe d'unités factices. (B) Le processus de communication entre les unités Faux. (C) Nous remarquons que l'un des chemins a été supprimé par connectivité mixte en raison de la multiplicité des chemins	23
Figure II.8: (a) Fenêtre d'échelle (3x3). (B)	24
Figure II.9: Représente le répertoire des blocs d'images	25
Figure II.10: L'analyseur de code explique.	26
Figure II.11: (a) Fenêtre d'échelle (3x3). (B) Une coupe transversale des unités d'image de la matrice d'image.....	27
Figure II.12. Regroupement LBG pour l'espace à 2 dimensions	29
Figure II.13: Processus de compressions.....	30

Chapitre III: Compression des images par l'Algorithmes LBG

Simulation Et Résultats

Figure III.1: Schéma programme en langage MATLAB.....	38
Figure III.2: Organigramme d'algorithm LBG	40
Figure III.3: Voir l'image au programme MATLAB.....	45
Figure III.4: extrait Méthodes (PCSM) et (CTM) au programme MATLAB	47
Figure III.5: Voir l'image compressée au programme MATLAB	48
Figure III.6: Voir l'image originale et l'image compressée au programme MATLAB	49

Liste des figures

Figure III.7: Voir (MSE) et (PSNR) au programme MATLAB.....	50
Figure III.8 Voir deux l'image originale et compressée FQ= 1	51
Figure III.9 Voir deux l'image originale et compressée FQ=3.....	52
Figure III.10 Voir deux l'image originale et compressée FQ=10	53
Figure III.11 Voir deux l'image originale et compressée FQ=20	54
Figure III.12 Voir deux l'image originale et compressée FQ=50	55
Figure III.13 Voir deux l'image originale et compressée FQ=80	56

LISTE DES TABLEAUX

Chapitre II: Techniques compression d'images par LBG

Table II -1: la comparaison entre l'existant et technique ... **3Error! Bookmark not defined.**

Chapitre III: Compression des images par l'Algorithmes LBG Simulation Et Résultats

<u>Tableau III.1: Un exemple de méthode de mutation</u>	<u>42</u>
<u>Tableau III.2: Exemple de méthode de mutation 2</u>	<u>43</u>
<u>Tableau III.3: Méthodes de signification transactionnelle progressive (PCSM).....</u>	<u>46</u>
<u>Tableau III.4: Méthodes de seuil transactionnel (CTM)</u>	<u>46</u>
<u>Tableau III.5 Résultats de la compression avec FQ=1</u>	<u>51</u>
<u>Tableau III.6 Résultats de la compression avec FQ=3.....</u>	<u>52</u>
<u>Tableau III.7 Résultats de la compression avec FQ=10.....</u>	<u>53</u>
<u>Tableau III.8 Résultats de la compression avec FQ=20.....</u>	<u>54</u>
<u>Tableau III.9 Résultats de la compression avec FQ=50.....</u>	<u>55</u>
<u>Tableau III.10 Résultats de la compression avec FQ=80.....</u>	<u>56</u>
<u>Tableau III.11 Résultats de la compression avec FQ= (1 jusqu'à 80).....</u>	<u>57</u>

Liste D'acronymes

LBG: Linde-Buzo-gray

PSNR : Peak signal-to-noise ratio

MSE : Mean square error

SNR : Signal-to-noise ratio

VQ: La quantification vectorielle

QS: quantification scalaire

CBIR: Content Based Image Retrieval

RVB: Red Green Blue en anglais

YCbCr: color spaces used as a part of the color image pipeline en anglais

PNN: probabilistic neural network

Introduction générale

Le besoin humain de représentations graphiques pour transmettre des connaissances ou pour enregistrer des situations dont on ne veut pas se souvenir. L'utilisation d'images est justifiée par le fait que le contenu présenté est facilement compréhensible par de larges catégories de personnes ayant une culture similaire.

La compression des images peut être obtenue en les transformant en les projetant sur la base de fonctions orthogonales, puis en encodant le résultat de la transformation. ont suscité beaucoup d'intérêt dans le traitement du signal. Ils permettent de représenter le signal avec un nombre fini de coefficients, la localisation exacte de la discontinuité étant déterminée.

Le but de ce mémoire est de compresser les images en niveaux de gris à l'aide d'un algorithme LBG et Codebook.

Notre mémoire est organisée selon les chapitres suivants :

Dans le premier chapitre : Nous étudierons les propriétés de l'image, les différents types, quelques concepts de base des images, et enfin les traitements qui peuvent leur être appliqués.

Dans le deuxième chapitre : Nous présenterons ci-dessous: algorithme LBG et codebook quelques concepts de base sur la compression d'image numérique sans perte et les mesures par lesquelles ses performances sont évaluées.

Dans le troisième chapitre : Nous présenterons l'organigramme ainsi que l'algorithme utilisé pour la compression ainsi que les différents résultats de simulation obtenus et analysés, avec MATLAB nous terminerons par une conclusion générale.

Chapitre I

Notions de traitement d'image

I.1 Introduction

Dans de nombreux domaines, l'image numérisée remplace les images analogiques classiques nécessaires pour la décrire. La durée de transmission est souvent trop longue et le stockage trop coûteux. Pour remédier à ces problèmes, il est donc nécessaire de réduire la taille de l'image en extrayant l'information visuelle utile qui sera la seule codée. Cette réduction de taille des données sans ou avec une perte non significative d'informations est effectuée par une technique de compression.

De ce fait, le terme générique traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes en regard de l'application concernée. [1] Avant de donner les notions fondamentales et les outils de traitement d'images, il serait utile de définir la notion d'image et les raisons de son traitement.

I.2 Définition d'une image

L'image est définie par :

- Le nombre de pixels qui la composent en largeur et en hauteur.
- L'étendue des teintes de gris ou des couleurs que peut prendre chaque pixel [2].

Elle peut être décrite sous la forme d'une fonction $I(x, y)$ de brillance analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine qui nécessite sa numérisation [3].

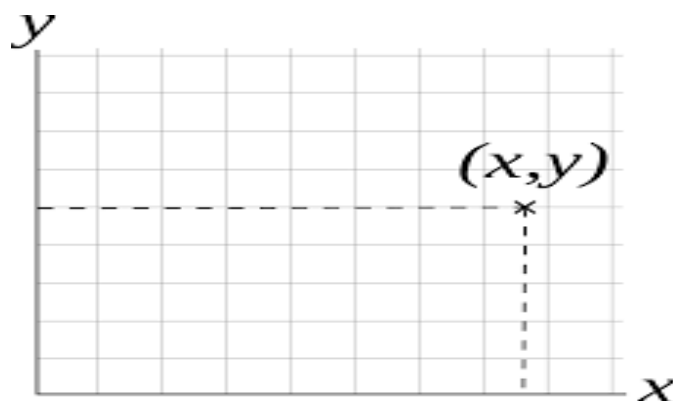


Figure I.1: Un point de l'image de coordonnées (x, y)

I.3 Notions fondamentales en traitement d'images

I.3.1 Pixel

Abréviation de « Picture Élément », mot anglo-américain de pix pour pics, abréviation de « Picture » (image) et d'el pour « élément ». Élément d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la Figure I.2 ci-dessous :

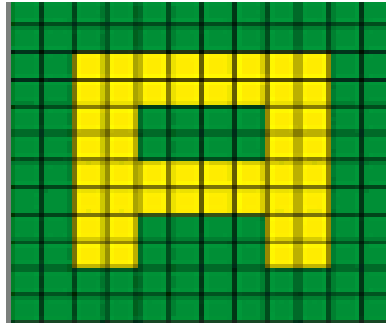


Figure I.2: Représentation de lettre A sous la forme d'un groupe de pixels

La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs. Dans le cas d'une image monochrome, chaque pixel est codé sur un octet, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image.

Dans une image couleur (R.V.B.), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B). [1.2]

I.3.2 Voisinage d'un pixel

Le voisinage d'un pixel est composé de tous les pixels qui l'entourent immédiatement. Si p est un pixel d'une image D , alors le voisinage de p est le plus petit sous-ensemble de D qui contient p .

Dans une image numérique, on distingue deux types de connexités relatives au voisinage utilisé : la 4-connexité (Figure I.3.a) et la 8-connexité (Figure I.3.b). Le voisinage d'un pixel (i,j) est dit 4-connexe s'il est formé des quatre pixels de coordonnées spatiales

$(i+1,j), (i,j+1), (i-1,j), (i,j-1)$, et il est dit 8-connexe s'il est formé des pixels de coordonnées spatiales $(i+1,j), (i,j+1), (i-1,j), (i,j-1), (i+1,j-1), (i+1,j+1), (i-1,j+1), (i-1,j-1)$ [3].



Figure I.3: Voisinage d'un pixel

I.3.3 Image numérique

Contrairement aux images obtenues à l'aide d'un appareil photo, ou dessinées sur du papier, les images manipulées par un ordinateur sont numériques (représentées par une série de bits).

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter.

La numérisation d'une image est la conversion de celle-ci de son état analogique (distribution continue d'intensités lumineuses dans un plan xOy en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques $f(x,y)$ où :

x, y : coordonnées cartésiennes d'un point de l'image.

$F(x, y)$: niveau de gris en ce point

Pour des raisons de commodité de représentation pour l'affichage et l'adressage, les données images sont généralement rangées sous formes de **Figure I.4** (matrice) de N lignes et P colonnes.

Chaque élément $I(x, y)$ représente un pixel de l'image et à sa valeur est associé à un niveau de gris codé sur m bits (2^m niveaux de gris ; $0 = \text{noir}$; $2^m - 1 = \text{blanc}$). La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur. [1]

$$I = \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,P-1) \\ f(1,0) & f(1,1) & & & f(1,P-1) \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ f(N-1,0) & \dots & & & f(N-1,P-1) \end{pmatrix}$$

Figure I.4: matrice

I.3.4 Résolution d'une image

La résolution d'une image est définie par un nombre de pixels par unité de longueur de la structure à numériser (classiquement en dpi (dots per inches) ou ppp (points par pouce)). Ce paramètre est défini lors de la numérisation et dépend principalement des caractéristiques du matériel utilisé lors de processus de numérisation. Plus le nombre de pixels est élevé par unité de longueur de la structure à numériser, plus la quantité

D'information qui décrit cette structure est importante et plus la résolution est élevée. La résolution d'une image numérique définit le degré de détail qui va être représenté sur cette image.

I.3.5 Poids d'une image

Pour connaître le poids (en octets) d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre de cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. Le poids d'image est alors égal à son nombre de pixels que multiplie le poids de chacun de ces éléments. [4]

Exemple : Calcul pour une image 640x480 Nombre de pixels :

$$640 \times 480 = 307200.$$

Le poids de l'image est ainsi égal à :

$$307200 \times 3 = 921600 \text{ octets.}$$

$$921600 / 1024 = 900 \text{ ko.}$$

I.3.6 Image en niveaux de gris

Dans une image en niveaux de gris, la couleur d'un pixel peut prendre des valeurs allant du noir au blanc, en passant par un nombre fini de niveaux intermédiaires. Dans une telle image les intensités du rouge, du vert, et du bleu de chaque pixel sont égales. En général, les images en niveaux de gris renferment 256 teintes de gris (image à 256 couleurs), simplement chacune de ces 256 couleurs est définie dans la gamme des gris. Par convention la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale). Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la " couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux. [1]

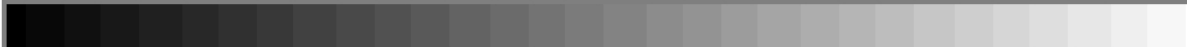
Valeurs des niveaux de gris et teintes de gris correspondantes
0....20....40....50....80....100....120....140....160....180....200....220....240....255


Figure I.5 : Valeurs des niveaux de gris et teintes de gris correspondantes.

Exemple :



Figure I.6: Exemple d'une image en niveaux de gris.

I.3.7 Adaptation à luminance et sa distinction

L'image se compose d'un certain nombre d'unités picturales, chaque unité picturale a une valeur numérique, qui détermine l'intensité d'éclairage de l'unité picturale, et ces valeurs numériques sont appelées niveau de gris, et comprises entre zéro, qui représente le noir, et le nombre (255), qui représente le blanc pur. Chaque unité picturale représente un niveau de gris spécifique indiquant l'intensité de la luminosité (**Brightness**) de cette unité.

Il est très important que l'œil puisse distinguer les niveaux de gris, puis distinguer les différents degrés de luminance, plus le degré de luminance est élevé, plus le niveau de gris est élevé. Le niveau de gris numéro 255 est représenté par (111111111), et le gris le niveau est zéro représenté par (00000000), et cela s'appelle (**Un octet**). Lorsque nous voulons représenter le seizième niveau de gris (16), nous n'avons besoin que de quatre chiffres (00001111) pour représenter ce niveau lorsqu'il est traité avec l'intensité d'éclairage (**Intensity**) dans le langage des nombres. Le niveau de gris (255) prend la valeur un (1) et le niveau de gris (0) la valeur zéro, et les niveaux restants vont de zéro (0) à un (1).

Nous notons que la relation entre l'intensité de l'éclairage entrant dans l'œil et la luminance perçue par le système visuel dans l'œil n'est pas une relation non linéaire, mais plutôt plus l'éclairage est élevé Par double, l'intensité de la luminance perçue par l'œil augmente dans une proportion constante "relation logarithmique" **Figure I.7** l'illustre [4].

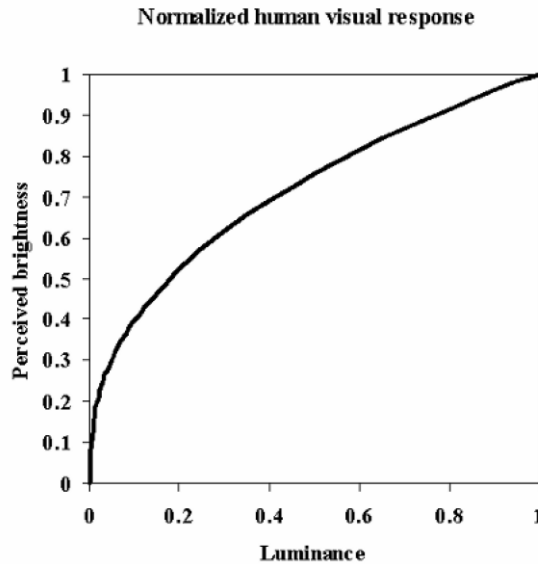


Figure I.7: Relation entre l'intensité de l'éclairage entrant dans l'œil et la luminance

Le phénomène le plus important est la capacité du système visuel humain à émerger ou à ressentir les zones dans lesquelles la valeur de l'intensité de la luminance change, afin de connaître avec précision et clarté le contenu de l'image. Il commence par la couleur noire foncée et se termine avec la couleur blanche brillante. Cette bande est appelée Mach Bands. Cette bande se compose de plusieurs couches, chacune avec une intensité fixe. [4]

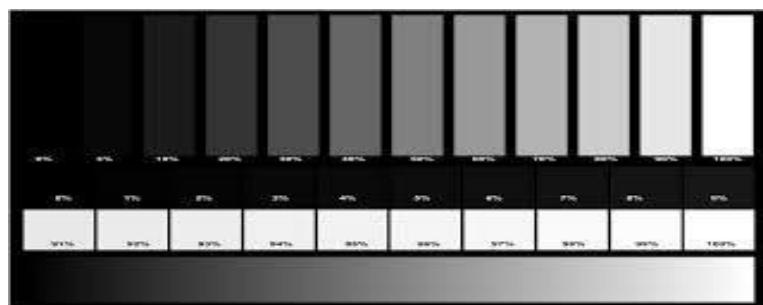


Figure I.8: Valeurs des niveaux de gris et teintes de gris correspondantes

Dans la **Figure I.9**, il n'y a que deux niveaux de gris, la couleur est noir foncé, et elle représente le niveau de gris le plus bas (0) (la luminance la plus basse), et la couleur blanche brillante, et elle représente le niveau de gris le plus élevé (255) (le plus lumineux).

D'après la figure, il est clair que la zone (A) est plus sombre et que la zone (B) est plus claire. [4]

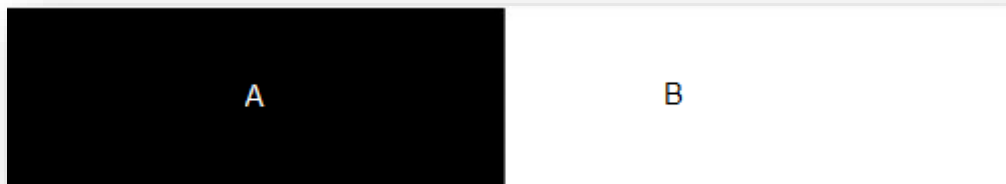


Figure I.9: montre la nette différence de niveaux de gris.

Grâce à la **Figure I.8** et à la Figure I.9, il devient clair pour nous que le sens du système visuel est plus grand dans la **Figure I.9** que dans la **Figure I.8**, et la raison en est due à la présence d'une grande différence Parmi les niveaux de gris de la **Figure I.8** et inclus dans les niveaux de gris de la **Figure I.9** La zone entourant l'objet visible "cible" est d'une grande importance pour la réponse de l'œil. Par exemple, la sensibilité du système visuel à une lumière petite et intense dans une pièce sombre est grande, et le sens du système visuel d'un objet représenté par le noir dans une pièce lumineuse l'est moins. La **Figure I.10** montre que la sensibilité du système visuel vis-à-vis du carré blanc sur fond noir est supérieure à celle vis-à-vis du carré noir sur fond blanc. Et nous avons tous les deux travaillé à augmenter le contraste entre le noir et le blanc, la sensibilité du système visuel à la forme a augmenté. [4]

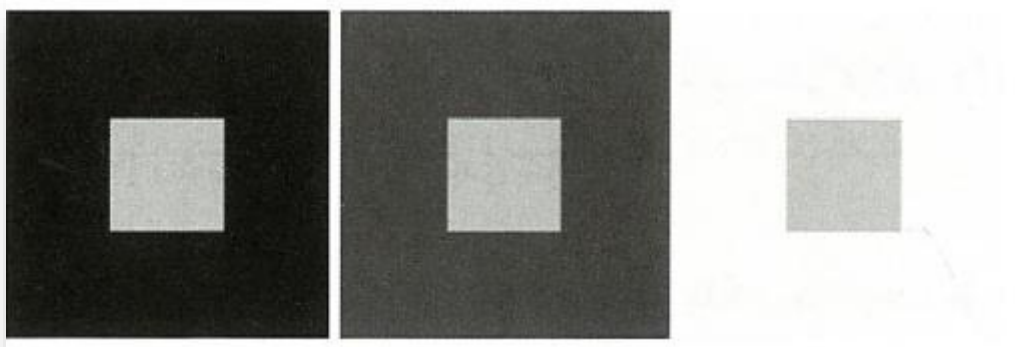


Figure I.10: Une explication du processus de contraste.

I.3.8 Image simple

Les images sont constituées de lumière réfléchiée par des objets, où $f(x, y)$ donne la valeur de luminance à ce point, et nous pouvons exprimer le point $f(x, y)$ avec deux composantes :

- La quantité de lumière tombant sur la vue que nous regardons et la représenter par $i(x, y)$
- La quantité de lumière émise par cette vue, représentée par $r(x, y)$

Grâce à des expressions mathématiques alors :

$$f(x, y) = i(x, y) r(x, y) \quad (0-1)$$

Comme la quantité de lumière incidente commence de zéro à l'infini.

$$\infty > i(x, y) > 0 \quad (0-2)$$

Nous constatons que la quantité de lumière réfléchiée par le corps varie entre zéro et un, et cela dépend de la qualité du matériau qui compose ce corps.

$$1 > r(x, y) > 0 \quad (0-3)$$

Et la valeur du niveau de gris au point (x, y) en diffère en un autre point $(x + 1, y + 1)$ et notons gray avec le symbole (L).

Tandis que:

$$L_{min} \leq L \leq L_{max} \quad (0-4)$$

$$r_{min} \times i_{min} \leq L \leq r_{man} \times i_{max} \quad (0-5)$$

$$0 \leq L \leq Level \quad (0-6)$$

I.3.9 L'image en couleurs

Une image en couleurs est censée représenter le mieux possible la réalité. La représentation des couleurs s'effectue de même manière que les images en niveaux de gris avec cependant quelque particularité. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composants primaires, le rouge, le vert et le bleu (le modèle R.V.B) [1] [9].



Figure I.11: Image numérisée avec 16 millions de couleurs.

Il y a des différentes représentations des images couleurs :

- La représentation en couleurs réelle sur 24 bits.
- La représentation en couleurs indexée ; on utilise une table appelée palette pour éviter la redondance de couleurs.
 - Une image RVB est composée de trois couches: Rouge, Vert et Bleu. Chaque pixel est défini par une valeur possible de ces couleurs de [0 à 255]. Une fois combinées, ces couches permettent de générer toutes nuances de couleurs [10].

I.3.10 Bruit dans l'image

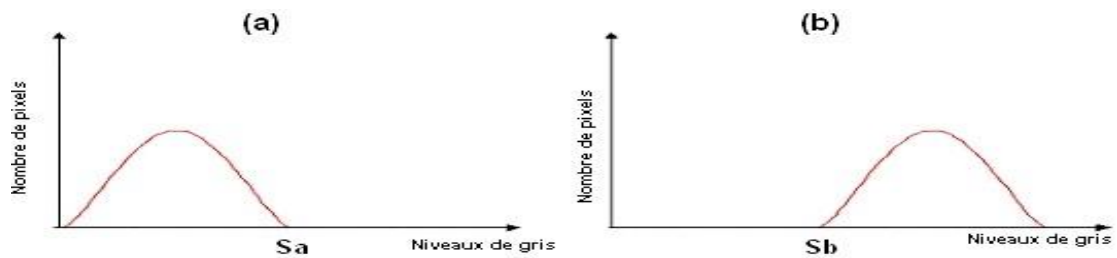
Le bruit dans une image est défini comme étant un phénomène de brusque variation d'un pixel isolé par rapport à ses voisins, il affecte la qualité de l'image et il provient soit du dispositif d'acquisition (scanner, caméra, amplification, quantification...) soit de la scène elle-même (poussières, rayures...).

I.3.11 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant. [1]

Il permet de donner un grand nombre d'informations sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée. (**Figure I.12**)

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



0

Figure I.12: Histogramme d'image.

I.4 Nature des images

Les images à compresser peuvent être de différentes natures [11] :

❖ Image photographiques

Ce sont généralement des images de scènes naturelles dans lesquelles l'intensité lumineuse varie de manière relativement continue.

❖ les images (modales)

Dans lesquelles l'intensité lumineuse est très changeante localement (dessin manuel par exemple). Leur histogramme est multimodal.

La numérisation des images peut aussi se faire de différentes manières, retrouvant :

- **Les images binaires** : qui contiennent seulement deux niveaux de gris différents.
- **Les images multi-niveaux** : numérisées sur plus d'un (1) bit par pixel.
- **Les images multicanaux** : dont l'exemple classique est celui des images couleurs.

I.5 Domaines d'applications du traitement d'images

Le traitement d'images possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que les télécommunications (TV, vidéo, publicité,...), la médecine (radiographie, ultrasons,...), la biologie, l'astronomie, la géologie, l'industrie (robotique, sécurité), la métrologie, l'architecture, l'imprimerie, l'armement (application militaire).

De nouvelles applications pratiques possibles aujourd'hui et qui touchent tous les domaines d'activités, tels que : les métiers du spectrale, de la radio, les créations artistiques [16].

I.6 Etape de l'analyse d'une image

Les étapes de traitement d'images sont:

- L'acquisition, l'échantillonnage, la quantification.
- L'analyse globale de l'image et transformations ponctuelles : Histogramme ; statistiques (moyenne, écart-type ...), transcodage (palette de couleur [LTU]) et classification.
- L'amélioration, le filtrage et la segmentation : Opération de convolution (lissage, rehaussement, détection de contours), squelettisation, vectorisation.
- Interprétation et sémantique : Cartographie thématique (classification automatique et supervisée), cartographie vecteur, représentation des graphes et de la topologie [15].

I.7 Conclusion

Dans ce chapitre, nous avons vu et étudié les différentes caractéristiques de l'image, précisément sa taille qui pose un problème dans sa capacité de stockage et dans sa transmission à travers les différents réseaux surtout en matière de vitesse.

Ainsi pour résoudre ce problème l'image doit passer par telles méthodes de compression dite sans ou avec perte, en cherchant d'une part, à réduire au maximum sa taille et d'autre part à grader, par la même occasion et au mieux possible, sa qualité.

Pour cela nous allons étudier dans le chapitre suivant, les techniques de compression par LBG.

Chapitre II

Techniques compression d'images par LBG

II.1 Introduction

La quantification vectorielle (**VQ**) est une approche efficace et simple pour la compression de données [2.1] [2.2][2.3]. Puisqu'il est simple et facile à mettre en œuvre, **VQ** a été largement utilisé dans différentes applications, telles que le modèle reconnaissance [4], détection de visage [5], segmentation d'image [2.6] [2.7] [8], compression de données vocales [9], Content Based Image Retrieval (**CBIR**) [10], dans les images [11][12] etc. La quantification vectorielle est une technique de compression d'image avec perte. Il existe trois procédures principales dans **VQ**, à savoir la génération codebook, la procédure de codage et la procédure de décodage. Dans le codebook processus de génération, l'image est divisée en plusieurs vecteurs d'entraînement k-dimensionnels. Le représentant codebook est généré à partir de ces vecteurs d'apprentissage par les techniques de regroupement. Dans l'encodage procédure, une image originale est divisée en plusieurs vecteurs k-dimensionnels et chaque vecteur est codé par l'index du mot de code par une méthode de consultation de table Les résultats codés sont appelés une table d'index. Au cours de la procédure de décodage, le récepteur utilise le même livre de codes pour retraduire l'index vers son mot de code correspondant pour reconstruire l'image. L'un des points clés de **VQ** est de générer un bon codebook [13] tel que la distorsion entre l'image originale et l'image reconstruite soit le minimum. Afin de trouver le mot de code le mieux adapté dans l'encodeur, le codage **VQ** ordinaire utilise l'algorithme de recherche complet, qui examine la distance euclidienne entre l'entrée vecteur et tous les mots de code dans le livre de codes. C'est un processus qui prend du temps. Pour surmonter cela, un jeune L'algorithme **LBG** de recherche est rapporté pour la compression d'image basée sur **VQ** [14] [15].

II.2 Définition de la compression

La compression d'image est une application de la compression des données sur des images numériques. Elle a pour utilité de réduire la redondance des données dans une image afin de pouvoir l'emmagasiner sans occuper beaucoup d'espace ou la transmettre rapidement. La compression d'images peut être effectuée avec perte de données ou sans perte [18].

Un compresseur utilise un algorithme qui sert à optimiser les données en utilisant des considérations propres au type de données à compresser.

Un décompresseur est donc nécessaire pour reconstruire les données originales grâce à l'algorithme inverse de celui utilisé pour la compression [19].

II.3 Principe général de la compression d'image

Le schéma général souvent utilisé pour décrire le fonctionnement des Algorithmes de compression est celui présenté dans la **Figure II.4**.

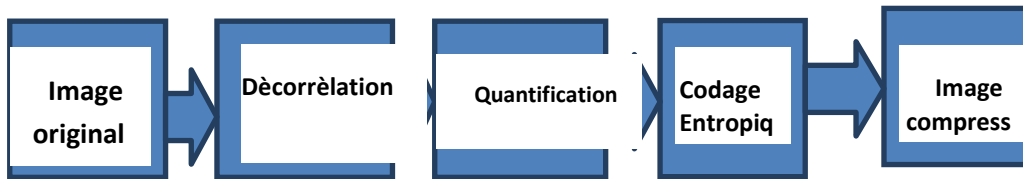


Figure II.1: Schéma synoptique de compression

❖ Décorrélacion (Transformation)

La dépendance existante entre chacun des pixels et ses voisins (la luminosité qui varie très peu d'un pixel à un pixel voisin) traduit une corrélation très forte dans l'image. Pour réduire le volume d'informations dans une image, l'opération de décorrélacion des pixels s'avère nécessaire. Elle consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés. C'est une opération réversible [20].

II.4 Les deux principes de base méthodes de compression

Les deux principes sur lesquels s'appuient les méthodes de compression sont l'extraction de l'information pertinente et l'élimination des redondances, au plan spatial comme au plan temporel. Ils sont utilisés conjointement dans un schéma complet de codage. Voici une liste synthétique des principales méthodes utilisées.

❖ *Extraction de l'information pertinente*

- **Seuillage.**
- **quantification des coefficients.**
- **quantification vectorielle.**
- **techniques de quantification adaptative.**
- **sous-échantillonnage spatial et temporel.**

❖ *Élimination des redondances*

- **Codage prédictif**
- **modulation Delta,**
- **MICD ("DPCM") ligne par ligne,**

- MIC 2-D ("DPCM 2-D"),
- prédiction intertrame.
- Décorrélation
- Transformée en Cosinus Discret (« Discrete Cosine Transform »),
- codage en sous-bande,
- codage par transformation 3-D.
- Codage entropique
- codage des longueurs de plage,
- codage de Huffman,
- codage par plans de bits.
- codage arithmétique

❖ Quantification

La quantification réduit le nombre de bits nécessaires pour le stockage des coefficients après la transformation par une dégradation de la précision de ces derniers. La quantification peut être appliquée sur des coefficients individuels (quantification scalaire QS) ou sur un ensemble de coefficients (quantification vectorielle Q [18]).

❖ Codage Entropique

Le codage entropique effectue un codage sans sur les valeurs quantifiées.

Cette dernière étape est nécessaire dans les méthodes sans perte, mais elle est souvent présente aussi dans les algorithmes irréversibles, puisque les valeurs transformées et quantifiées contiennent davantage de redondances. Cependant, l'absence du codeur entropique peut être justifiée et nécessaire, à cause notamment des contraintes de transmission [21].

II.5 Echantillons et quantification

II.5.1 Echantillons et quantification régulière

Pour que l'image soit apte au traitement informatique, il faut représenter ces images avec des informations (**Data**) qui correspondent au contenu de ces images, car l'image est numérotée (**Digitized**) dans chacun des espaces (**Spatial**), et nous appelons cela en numérisant les échantillons d'image (**Image**) (**Sampling**, et la

largeur (**Amplitude**) et nous appelons cela la quantification des niveaux de gris **Gray Level** (**Quantization**) [23].

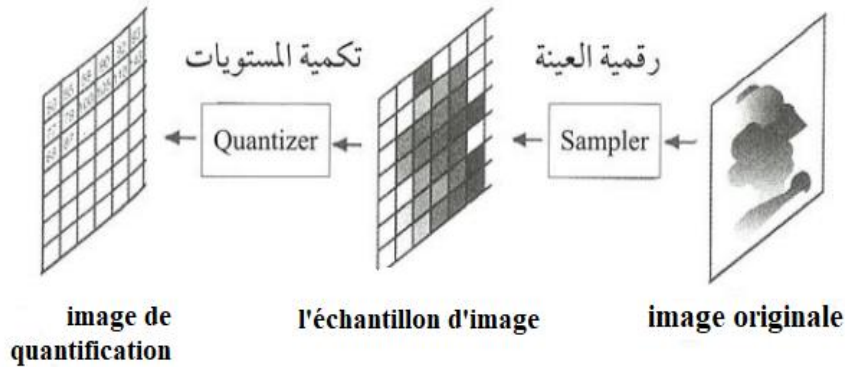


Figure II.2: Une explication du processus de contraste.

Lorsque nous avons une image représentée par un tableau comme : $N \times M$

$$f(x, y) = \begin{bmatrix} f(0.0) & f(0.1) & \dots & f(0.M-1) \\ f(1.0) & f(1.1) & \dots & f(1.M-1) \\ \vdots & & & \\ f(N-1) & f(N-1.1) & \dots & f(N-1.M-1) \end{bmatrix} \quad (0-7)$$

Les modules peuvent être réguliers, disposés et régulièrement espacés, comme indiqué **Figure II.3.A)** et les unités factices peuvent être irrégulières, comme le montre la **Figure II.3. B)** [6].

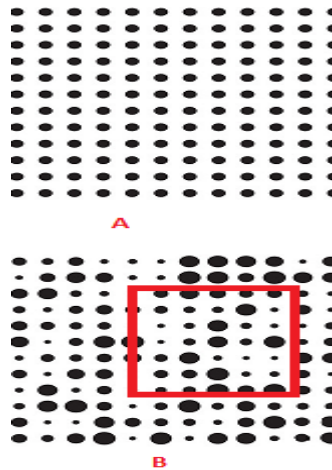


Figure II.3: (A) Unités factices régulièrement espacées. (B) Unités picturales irrégulières.

II.5.2 Nombre d'une unité image

Une des choses les plus importantes que nous devons savoir est le nombre d'unités d'image avec lesquelles nous devons représenter l'image et connaître le nombre de niveaux de gris pour chaque unité d'image de cette image afin que nous puissions connaître l'espace dont cette image a besoin pour le stockage. Il est de notoriété publique dans les systèmes de traitement d'images numériques que le niveau de gris peut être connu. Ce dont chaque unité visuelle a besoin à travers l'équation suivante :

$$G = 2^m \quad (0-8)$$

G: Le niveau de gris de l'unité graphique.

M : le nombre de (bits) qui représentent le niveau de gris de l'unité graphique

Pour obtenir le nombre de bits dont vous avez besoin pour stocker un tableau d'images contenant $N \times M$ et représenté par un niveau de gris contenant le nombre de bits, l'équation suivante s'applique :

$$b = N \times M \times m \quad (II-9)$$

II.6. Quelques relations de base entre les unités picturales

II.6.1 Faux voisins d'unité d'image.

L'image est un ensemble d'unités d'image. Chaque unité factice est entourée d'un groupe d'unités factices dans toutes les directions. **Figure II.7** [25].

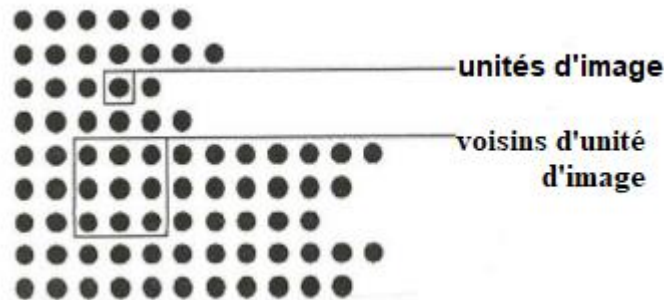


Figure II.4: Unités d'image et ses voisins

Prenons une unité d'image et notons-la avec le symbole (p), et elle est à une coordonnée (coordonnée) connue, et (x, y) sa position dans l'image. Cette unité visuelle est entourée d'unités horizontales (Horizontales) et verticales (verticales), comme le montre la **Figure II.8** [21].

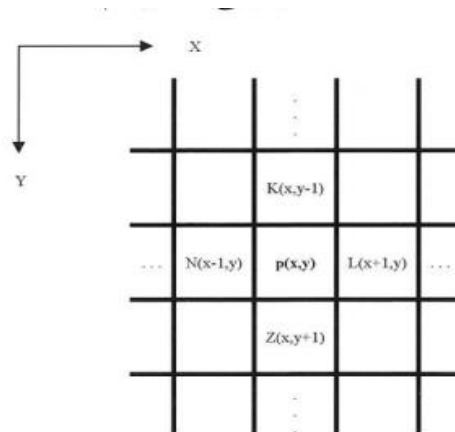


Figure II.5: Unité d'image Nominal $m(x, y)$ Dans une image, les quatre voisins par position de coordonnées.

Ces unités sont appelées les quatre voisines (4-Neighbors) de l'unité (p) et sont désignées par le symbole $N_4(p)$ Il existe d'autres voisins de l'unité (p), et ils sont appelés voisins diagonaux (4-diagonal) et ils sont représentés sur la **Figure II.9** et

nous les désignerons par le symbole $N_D(p)$. Toutes les unités horizontales, verticales et diagonales précédentes sont appelées les huit voisins (8-Neighbors) et sont désignées par le symbole $N_8(p)$. Lorsque l'unité (p) est sur un bord de l'image, certaines unités adjacentes à cette unité seront manquantes (à l'extérieur de l'image).

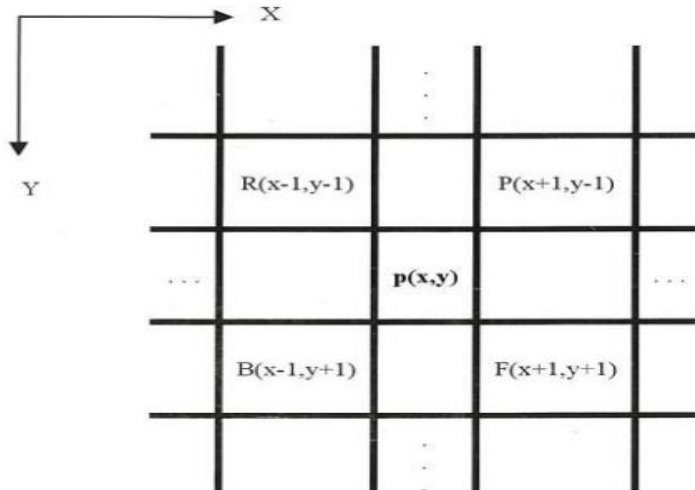


Figure II.6: Unité d'image $p(x, y)$ dans une image et les pays voisins répartis en fonction de l'emplacement des coordonnées

II.6.2. Conductivité

La connectivité est une relation entre les unités d'image, et c'est un concept important que nous utilisons pour définir et clarifier les limites des objets et des formes dans une image et les composants de différentes régions. Lorsque nous appelons le terme communicant à deux unités de l'image, nous devons déterminer si ces deux unités étaient adjacentes ou non ?

Et s'ils le sont, sont-ils égaux dans l'échelle de niveaux de gris qui représente chaque unité. Il devient clair au bord du visage qu'il n'y a pas de continuité avec la figure postérieure, et la raison

En est une différence d'échelle de gris entre les unités graphiques, qui représentent le visage, et les unités graphiques, qui représentent l'arrière-plan.

Il existe plusieurs types de communication, qui sont les suivants :

- ✓ Connectivité quaternaire (4-Connectivity) : On peut dire : L'unité nominale (q) est reliée à l'unité (p) si (q) est dans le groupe $N_4(p)$.
- ✓ La huit connectivité (8-Connectivity) : unité (q) et (p) sont connectés Huit si (q) est dans le groupe $N_8(p)$.
- ✓ Connectivité mixte (Mixed-Connectivity) : On peut dire que l'unité (q) et l'unité (p) sont connectées mixtes si :
 - est dans le groupe $N_4(p)$ ou
 - est dans le groupe $N_D(p)$ et groupe $N_4(p) \cap N_4(p)$ Vide.

La connectivité mixte n'est rien d'autre qu'une modification de la deuxième connectivité, créée pour éliminer les chemins multiples. La **Figure II.10 A** montre un groupe de voisins des unités fictives, et la **Figure II.10 B** une illustration de la deuxième connexion et la **Figure II.10 C** montre la connectivité mixte [1].

II.6.3 Opérations arithmétiques

Les opérations arithmétiques et logiques que nous utiliserons entre les unités d'image sont importantes dans le processus de traitement d'image. Les opérations arithmétiques sont appliquées aux unités fictives.

Si nous avons une unité fictive (p) avec une unité fictive (q), alors nous pouvons utiliser les formules arithmétiques suivantes :

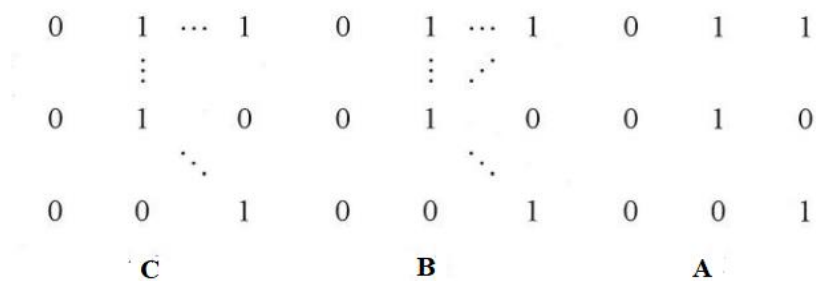


Figure II.7: (A) Un groupe d'unités factives. (B) Le processus de communication entre les unités Faux. (C) Nous remarquons que l'un des chemins a été supprimé par connectivité mixte en raison de la multiplicité des chemins.

(Binary) tandis que les opérations arithmétiques sont appliquées aux unités d'image multi values toutes les opérations précédentes ; L'"arithmétique" et la "logique" que nous utilisons pour traiter l'image sur deux bases. La première base est qu'une unité fictive est traitée, et celle qui la suit à partir d'autres unités fictives, comme le

processus de combinaison de deux images est combinée sur la base d'une unité fictive avec l'autre unité fictive correspondante tandis que la deuxième base est le traitement par la méthode de voisinage Déterminé (Neighbors), et les processus de fenêtre (Window Operations) sont utilisés. Tels que les masques (Mask) et les filtres (Filters). La **Figure II.11A** montre une fenêtre (3X3). Le but de l'utilisation des opérations de fenêtre est de rendre la valeur de niveau de gris attribuée à l'unité d'image liée et affectée par les valeurs de

Les opérations booléennes sont appliquées aux images représentées par des nombres binaires, niveaux de gris dans les unités d'image adjacentes, de sorte que nous ayons un tableau d'une image, et nous avons pris un segment de elle qui constitue un certain nombre d'unités d'image, comme sur la **Figure II.11 B** [21].

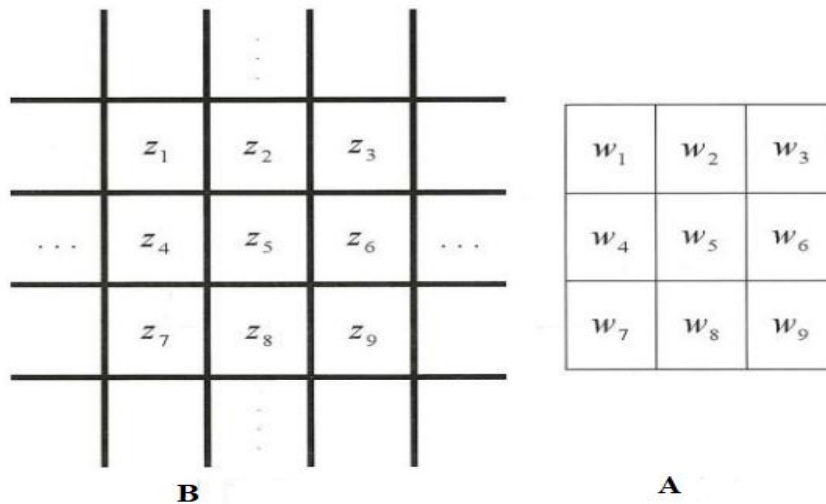


Figure II.8: (a) Fenêtre d'échelle (3x3). (B) Une coupe transversale des unités d'image de la matrice d'image

Nous en avons besoin. Supposons que nous souhaitons remplacer la valeur de niveau de gris du simulacre en effectuant les calculs suivants :

\bar{e} : La nouvelle valeur du niveau de gris de l'unité d'image z_5 .

w_j : Paramètres de la fenêtre.

z_i : Valeurs de niveaux de gris qui représentent des unités fictives.

Exemple n° (2). Soit une image contenant (512 x 512) unités image, et représentée par (128) niveaux de gris. Ce qu'il faut pour stocker cette image est:

$$G = 2^m \quad (0-10)$$

$$128 = 2^m \quad (0-11)$$

$$m = 7 \rightarrow \ln(128) = m \ln(2) \quad (0-12)$$

En appliquant l'équation 0-13, il en résulte que :

$$b = 512 \times 512 \times 7 = 1835008 \quad (0-13)$$

II.7. La quantification vectorielle et codebook

Sous forme de points (**pixl**) le répertoire de blocs d'images qui est stocké dans un index spécifié dont la taille est plus petite que le vecteur d'origine afin que ce vecteur réalise la moindre distorsion de l'image pour encoder un certain nombre de paquets (**encoder**) ou de signal, l'encodeur (**blocs**) est utilisé dans l'état de transmission comme indiqué. **Figure II.1** montre les mécanismes de génération de blocs d'images dans le cas de la transmission et de la réception, le décodeur, et dans le cas de la réception, il y a un décodeur (**decoder**) utilisant les images de guidage des blocs d'images et les vecteurs d'origine [022] [021]

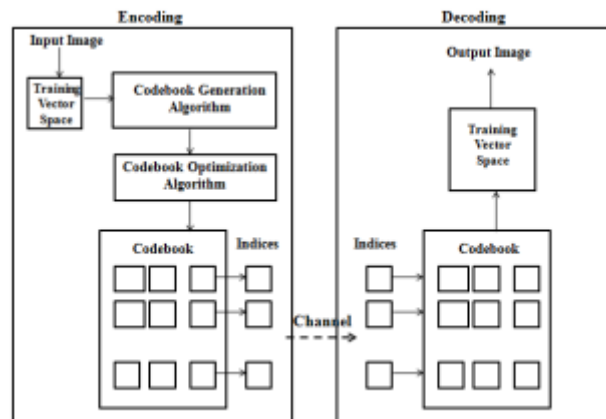


Figure II.9: Représente le répertoire des blocs d'images

Le guide des blocs d'images est utilisé dans plusieurs utilisations dans le domaine de Traitement des images, y compris en compression ou référence d'image, ainsi qu'en L'état de conservation des données transmises et dans certains cas dans Processus d'amélioration de l'image. Le processus de création (**construction**) du répertoire des blocs d'images est un processus de cryptage Une image ou un signe (**Encoding**) où les images sont divisées en Un groupe de blocs (**blocks**) ou vecteurs et chacun de ceux-ci Vecteurs d'indices spécifiques (**index**), **Figure II.2** illustrant Action de l'encodeur :

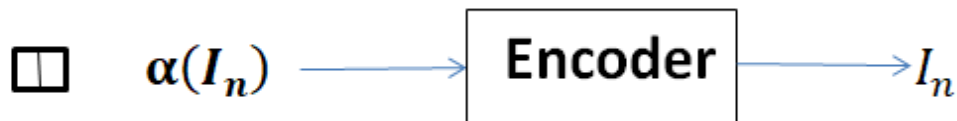


Figure II.10: Processus de codage.

La taille du bloc dépend du mot crypté de l'image Pour être quantifié et lors de la recherche de mots cryptés et de leur comparaison avec la valeur de L'image doit présenter le moins de distorsion possible Encodage L'équation **0-1** conduit au

processus d'organisation des entrées pour l'image (A_x) et effectuer une opération d'appariement (mapping) à l'ensemble Nouveau Z :

$$\alpha: A_x \rightarrow Z\{0,1\} \quad (0-1)$$

Où:

A_x : Les vecteurs de l'image d'origine sont classés par ordre alphabétique.

z : Les entiers qui seront envoyés A store Z : est un ensemble de via le canal de transport (**channel**).

Code parser (**decoding**) est utilisé pour extraire le bloc correspondant pour l'index demandé qui correspond au mot de code indexé dans le répertoire avec Le mot indexé reçu via le vecteur canal puis le vecteur est récupéré Original du manuel comme indiqué **Figure II.3**.

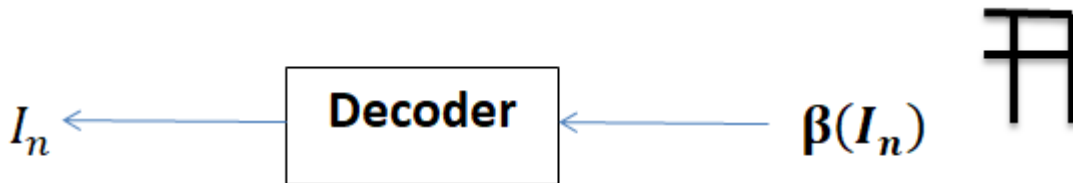


Figure II.11: L'analyseur de code explique

Où le cryptanalyste (β) définit :

$$\beta : Z \rightarrow \hat{A} x \quad (0-2)$$

Où $\hat{A}x$ est la valeur à régénérer et est généralement égal à A l'original

$$C = \beta(i); i \in Z \quad (0-3)$$

Où C est qui reproduit l'image dont la valeur est :

$$C = \{\beta i; i = 1, 2, 3 \dots \dots \dots M - 1 \quad (0-4)$$

Le code presse est A pour la source $\{X_n\}$ De ce qui précède, il ressort que la lame de passage Il contient les trois éléments (α, β, Z) comme suit:

$$A \xrightarrow{\alpha} Z \xrightarrow{\beta} C \quad (0-5)$$

$\hat{A}x$: calcule le mot de code le plus proche de l'image lorsque la quantité de distorsion ($d(x, y)$) est Moins que les autres mots du guide. De ce qui précède, il a été montré Il est clair que l'index de bloc schématique est un processus de

chiffrement Le répertoire se compose d'un certain nombre de mots de code (word code) qui Ils sont stockés sous forme de vecteurs (Vectors) et pour chacun d'entre eux Vecteurs Un index dont la taille doit être inférieure à la taille de Le vecteur d'origine, et il y a une échelle pour calculer la déformation $d(\hat{x}, x)$ entre Le vecteur d'image d'origine (x) (et le vecteur d'encodeur après avoir appuyé sur (\hat{x}) selon La loi indiquée dans l'équation **0-6**

$$d(x, \hat{x}) = \frac{1}{N}E(x, \hat{x}) \quad (0-6)$$

Où:

N : Le nombre de vecteurs.

X : Vecteurs de l'image originale

\hat{X} : Vecteurs d'images composites.

Le processus de réduction de répétition obtient les vecteurs codés qui sont

Comparez-le avec les vecteurs d'image d'origine et la taille de l'index est inférieure à taille du vecteur. Pour cette raison, l'indice de masse pictural est d'une grande importance dans Le quantificateur directionnel (Quantization Vector) La phase de formation Le guide passe par deux phases :

- La première étape: la formation de la preuve initiale (codebook Initial).
- La deuxième étape : la formation du guide idéal (codebook Optimal) Afin d'assurer des résultats efficaces, il existe plusieurs méthodes Il est utilisé
- pour choisir des vecteurs séquentiels, et ce choix est fait sur la base de Une échelle de filtre pour les valeurs aberrantes prenant en compte la distorsion.

II.8. CODEBOOK GENERATION ALGORITHMS

II.8.1 Linde, Buzo and Gray (LBG) Algorithm

L'algorithme de Linde-Buzo-gray (LBG) [2] est l'Algorithme de génération de codebook. Tout d'abord, l'image est divisée en blocs non superposés de taille fixe pour former le cluster initial. Ensuite, le centroïde du cluster initial est calculé. Après cela, ajoutez et soustrayez l'erreur constante de ce centroïde pour obtenir deux vecteurs d'erreur e_1 et e_2 pour les clusters 1 et 2 respectivement. Comparez chaque vecteur de cluster avec les vecteurs d'erreur e_1 et e_2 et diviser le cluster en deux. Pour la prochaine itération, calculez le centroïde du cluster et le calcul des vecteurs d'erreur pour diviser davantage ce groupe en deux. Répétez la procédure ci-dessus jusqu'à ce qu'elle soit donnée nombre de grappes ne se forment pas. Chaque vecteur de livre de codes est obtenu en calculant le centroïde de chaque cluster. La **Figure II.12** montre Regroupement LBG pour un espace à deux dimensions.

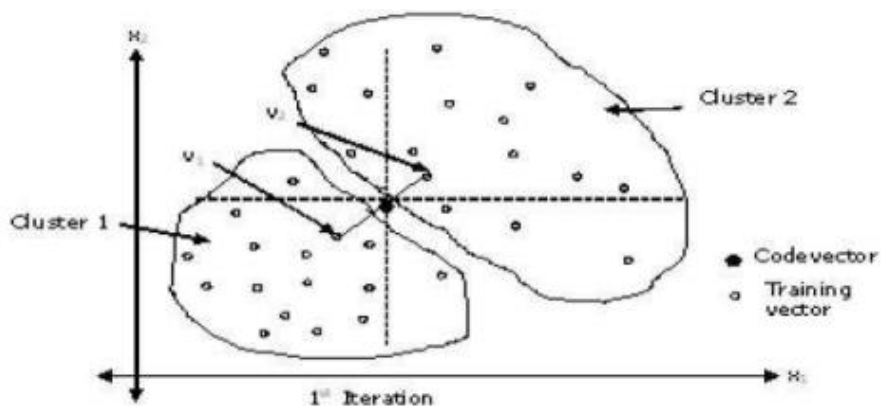


Figure II.12. Regroupement LBG pour l'espace à 2 dimensions

II.8.2 Linde-Buzo-Gray algorithm Convert RGB image

- Convertir l'image RVB en format de couleur YCbCr.
- Prenez uniquement le composant Chrominance Red (Cr) d'Image YCbCr, qui sera l'image d'entrée.
- L'image d'entrée de taille NXN est divisée en sous-blocs de taille n x n.
- Définir la taille du livre de codes et aléatoirement sélectionnez M vecteurs parmi les vecteurs d'entrée.

$$D(X, C) = \sqrt{\sum_{t=1}^k (X_t - C_t)^2} \quad (0-14)$$

Recherchez le vecteur de code le plus proche de C, puis ajoutez le vecteur d'entrée dans le groupe correspondant de le vecteur de code le plus proche trouvé.

- Pour chaque vecteur de code dans le livre de codes actuel C, trouvez le centre de gravité de son cluster associé et prenez le centre de gravité comme nouveau vecteur de code pour la prochaine itération. Répétez les étapes ' et , jusqu'à ce que le code Vector ne le fasse pas le changement ou le changement est petit.

- Mesurer les paramètres de performance PSNR et MSE de l'image RVB reconstruite. Le diagramme du processus de compression est illustré à la **Figure II.13**. L'image de couverture est convertie en couleur YCbCr l'espace, puis l'algorithme LBG est appliqué et le codebook et l'index est obtenu. Ainsi la couverture l'image est compressée.

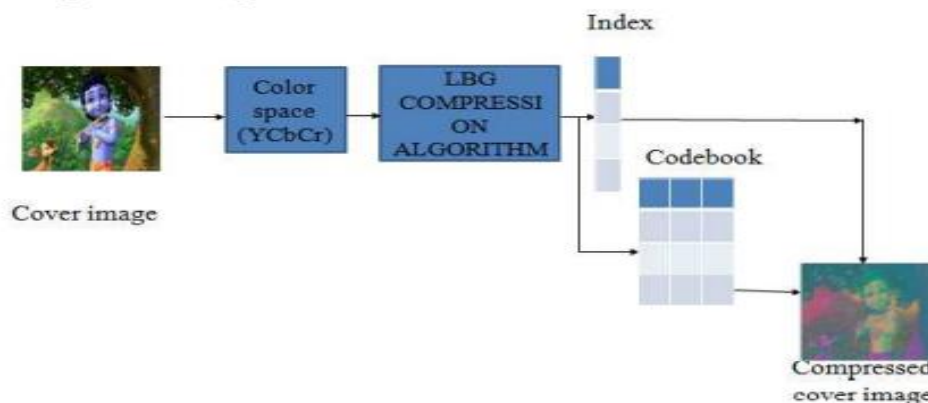


Figure II.13: Processus de compressions

II.8.3 Phase d'extraction

En phase d'extraction, l'image secrète est extraite de l'image de couverture compressée. Les étapes d'extraction des informations secrètes sont répertoriées comme suit :

- Lire stego-image puis décompresser la couverture image utilisant un livre de codes et un vecteur d'index.
- Appliquer IWT pour extraire l'image secrète de la couverture Image. Le tableau II-1 montre la comparaison entre l'existant et technique proposée :

Table 0-1: la comparaison entre l'existant et technique

Techniques	Couver image size	PSNR value for hiding image (in dB)	PSNR value forHiding text (indB)	SSIM
Existing	512x512 JPEG	48.067	50.543	0.64
2LSB embedding	512x512 JPEG	51.642	53.864	0.75
Proposed method:	512x512 JPEG	54.860	55.901	0.84

Le rapport signal/bruit maximal est la mesure de la qualité de l'image. Une bonne image doit posséder un faible MSE et un PSNR élevé.

II.9 Équations et éléments d'algorithme (LBG)

II.9.1 Équations

Le quantificateur directionnel est l'une des techniques efficaces qui sont utilisées dans De larges champs de traitement du signal et de l'image sont également utilisés dans la compression Images ou lorsque vous effectuez un processus de protection des fichiers en les chiffrant parce qu'ils sont basé sur le processus d'appariement (Q) (Euclidean Space) Multidimensionnel ($d_2(I, C_i)$) Ensemble partiel (Nc) À partir de ce l'espace et peut être défini par l'équation **0-15**

$$Q: d^2 \sum_{j=1}^k (I_j, C_{i,j}) \dots \dots i = 1, 2, \dots \dots N \quad (0-25)$$

Où:

I : représente l'image d'origine.

C_i : Le vecteur dans un tableau appelé indice schématique.

i : Représente le nombre de vecteurs dans l'index.

J : Éléments vectoriels représentant $K = (n * n)$.

Nc : la taille de l'index du bloc image.

La meilleure correspondance du vecteur avec la distorsion minimale est choisie,

Lequel s'appelle alors le gagnant, et peut être déterminé simplement par Équation (0-16):

$$d2(I, C_w) \text{ mind}^2(I, C) i = 1, 2, \dots \dots N \quad (0-16)$$

Où:

(Cw) Est le vecteur gagnant.

(W) : Représente la suite des vecteurs gagnants

Ce processus s'appelle la recherche complète (search Full) et se trouve un mécanisme de travail Le quantificateur directionnel comme le montre la *Figure II.1* pour diviser l'image en blocs A partir des dimensions, il est ensuite transformé en vecteurs singuliers appelés vecteurs d'entrée. Le code de chaque vecteur d'entrée est ensuite mis en correspondance avec le mot-clé d'inde

Les blocs d'image et l'obtention du meilleur vecteur d'entrée se font par Mesurez la distance euclidienne entre le vecteur d'entrée et le mot-clé comme indiqué par équation (0-17):

$$d(x, y_k) \leq d(x, y_j) \quad (0-17)$$

Pour tout $j = 1, 2, 3 \dots \dots \dots n$

$(d(X, Y))$ Représente l'échelle de distance, qui représente la déformation de Remplacement du vecteur d'origine (X) par le vecteur correspondant (Y).

L'encodeur est utilisé (dans le processus d'encodage de l'image lors de l'envoi) depuis Lors de l'insertion de chaque bloc (vecteur) de l'image dans le répertoire et le met L'indice (Index) est précisé, comme pour l'analyseur de code, qui se trouve à la fin L'autre extrait le numéro indexé du vecteur du

Répertoire et en génère Les vecteurs sortants et l'indice de masse schématique sont un problème important.

En (VQ), il existe de nombreux algorithmes utilisés pour composer le guide, dont les plus importants sont L'algorithme (Gray Buzz Lindo LBG) qui a été adopté dans ce PNN (Pair wise Nearest Neighbor) et algorithme, recherche Qui dépend du principe de voisinage et il existe un autre algorithme tel que Simulated Annealing Et Algorithme Maximum Desert C'est l'un des concepts de base qui se répètent dans les utilisations du quantificateur directionnel [2].

II.9.2 Éléments d'algorithme avec (LBG)

- La gastrectomie à manchon unique (Quantization Scalar) est un processus de quantification Un échantillon pendant une période de temps limitée.
- La quantification directionnelle (Quantization Vector) est un processus de quantification Une masse d'échantillons pendant une période de temps limitée et est aussi Suivant:
 - Un vecteur d'échantillonnage unidimensionnel est efficace lorsqu'il est utilisé avec, Le son ou la parole.
 - Un bloc d'échantillons se présente sous la forme d'une matrice à deux dimensions utilisée Avec compétence avec l'image.
- Pourcentage de distorsion (rate distortion) Le pourcentage de distorsion est calculé pour Obtenir une déformation avec un faible taux d'erreur ainsi que l'obtention de donne Le même rapport lors de l'utilisation du singulier (Scalar) du vecteur à la place de (Vector).

Encodeur : envoie la meilleure correspondance pour l'indexation à l'analyseur de chiffrement (Decoder).

L'analyseur de code utilise le répertoire comme liste de recherche (table Look up) Afin de réinstaller l'image ou la référence.

- Le magasin de l'encodeur (Buffer Encoder) est le bloc d'échantillons à chiffrer Dans lequel la meilleure correspondance est recherchée dans (le répertoire des blocs d'images Moins d'erreur carrée) Ordinaire
- L'indice de masse factice primaire (codebook Initial) est le guide qui Il est construit au début de l'algorithme afin d'obtenir la preuve finale.

II.10 Les avantages et Les inconvénients

Le répertoire idéal du bloc image : C'est le répertoire qui est généré après la procédure Améliorations finales des preuves primaires, qui dépendent en fin de compte par l'algorithme.

❖ Les avantages sont :

1) Il est possible d'étendre l'adoption de composants vectoriels de manière linéaire ou linéaire.

2) Simplicité de décodage par un analyseur de code qui repose uniquement sur Recherche dans la liste des répertoires de blocs d'images (table up Look) Ce qui nécessite un traitement informatique simple.

3) Fournit un rapport de codage fixe dans le cas de l'utilisation d'un quantificateur directionnel conventionnel.

4) Facilité de compatibilité avec d'autres codes de technologie de presse à des fins de production hybride Avec l'effet le plus élevé, un exemple de ceci est la transformation (quantifiée vectorielle)(VQ).

❖ augmente considérablement malédiction.

❖ 2) Le guide directionnel occupe un espace mémoire stocké et cet espace augmente Avec l'ajout de nouveaux blocs (vecteurs) au répertoire.

❖ 3) Le processus de création de blocs d'images nécessite un traitement important, en particulier dans Les dimensions sont importantes, ce qui n'est pas approprié dans le cas d'une transmission à l'heure Le vrai Lorsque l'encodage de cette image est différent des images qui (online).

❖ 4) Les performances ne sont pas élevées lors de l'encodage des images qui sont différentes des images Il est adopté lors d'entraînements à haute performance [3]. Quant à l'algorithme LBG, il dépend du processus d'exécution d'un processus de formation sur Un ensemble de chaînes pour l'image ou pour indiquer en supposant que le est divisé L'image est sur un nombre spécifié de chaînes de sorte qu'elle n'augmente pas et est calculée pour elle La quantité de distorsion d'image indiquée par l'équation *** doit dépasser une très petite valeur.

❖ Les inconvénients sont :

1) La méthode est lente à traiter et concentrée si le degré de sa complexité

$$D_{ave} = \frac{1}{mk} \sum_{m=1}^m \|x - c_1\|^2 \quad (0-18)$$

Dans ce cas, on obtient un indice de masse schématique initial, et pour obtenir Répertoire du bloc factice final Le hachage de ces chaînes se poursuit La quantité de déformation est calculée pour chacune des étapes de segmentation en

Les comparant Avec la quantité de déformation initiale qui souvent Comme le montre l'équation (0-19):

$$D_{ave}^1 = \frac{1}{mk} \sum_{m=1}^m \|x_m - Q(x_m)\|^2 \quad (0-19)$$

II.11 CONCLUSIONS

Dans cet article, les images secrètes peuvent être extraites sans toute distorsion de l'image d'origine. Cette technique **LBG** algorithm offre la haute qualité de l'image **Stego** ayant des valeurs **PSNR** élevées par rapport aux autres méthodes. Cependant, cette méthode peut nous permettre de transmettre les informations secrètes au destinataire indépendamment, il est presque impossible pour tout accident parties pour extraire les informations secrètes et récupérer l'image hôte d'origine, lorsque l'image Stego est accessible par leur **codebook**.

Chapitre III

Compression des images par la Algorithmes LBG

Simulation Et Résultats

III.1. Introduction

Notre système proposé utilise une quantification d'image en deux étapes basée sur un algorithme génétique statique (LBG) pour la compression d'image, et le but de la première étape est de générer le livre de codes initial utilisé pour classer les blocs d'image en fonction de certaines mesures de distorsion (Codebook) Alors que le script générera le meilleur bloc rejeté qui représente tous les blocs de chaque unité de code, dans la deuxième étape. D'après les résultats obtenus dans nos expériences, nous pensons que la solution au problème de quantification d'image est très prometteuse.

La recherche vise à étudier la possibilité d'obtenir des preuves de masses Une image idéale qui réalise le moins d'opérations et de temps passés Le processus de cryptage et de décodage a été adopté (LBG) algorithme commun Utiliser dans le quantificateur directionnel en l'appliquant à des exemples picturaux Ils sont cryptés, puis comparés et récupérés par des moyens qui servent le même objectif.

III.2. introduction au programme MATLAB

MATLAB est un logiciel de calcul matriciel à syntaxe simple. Avec ses fonctions spécialisées, **MATLAB** peut aussi être considéré comme un langage de programmation adapté pour les problèmes scientifiques. est un interpréteur : les instructions sont interprétées et exécutées ligne par ligne. fonctionne dans plusieurs environnements tels que **X-Windows**, Windows, Macintosh.

- ❖ Il existe deux modes de fonctionnement :
 - ✓ mode interactif : MATLAB exécute les instructions au fur et à mesure qu'elles sont données par l'utilisateur.
 - ✓ mode exécutif : MATLAB exécute ligne par ligne un "fichier M" (programme en langage MATLAB).

Fenêtre Commande : Dans cette fenêtre, l'utilisateur donne les instructions et MATLAB retourne les résultats.

Fenêtres Graphiques : MATLAB trace les graphiques dans ces fenêtres.

Fichiers M : Ce sont des programmes en langage MATLAB (écrits par l'utilisateur).

Boîtes à outils : Ce sont des collections de fichiers M développés pour des domaines d'application spécifiques (Signal Processing Toolbox, System

Identification Toolbox, Control System Toolbox, u-Synthesis and Analysis Toolbox , Robust Control Toolbox, Optimization Toolbox, Neural Network Toolbox , Spline Toolbox , Boîte à outils de chimométrie, Boîte à outils de logique floue, etc.)

Simulink : C'est l'extension graphique de MATLAB permettant de travailler avec des diagrammes en blocs.

Blocksets : Ce sont des collections de blocs Simulink développés pour des domaines d'application spécifiques (DSP Blockset, Power System Blockset, etc.).

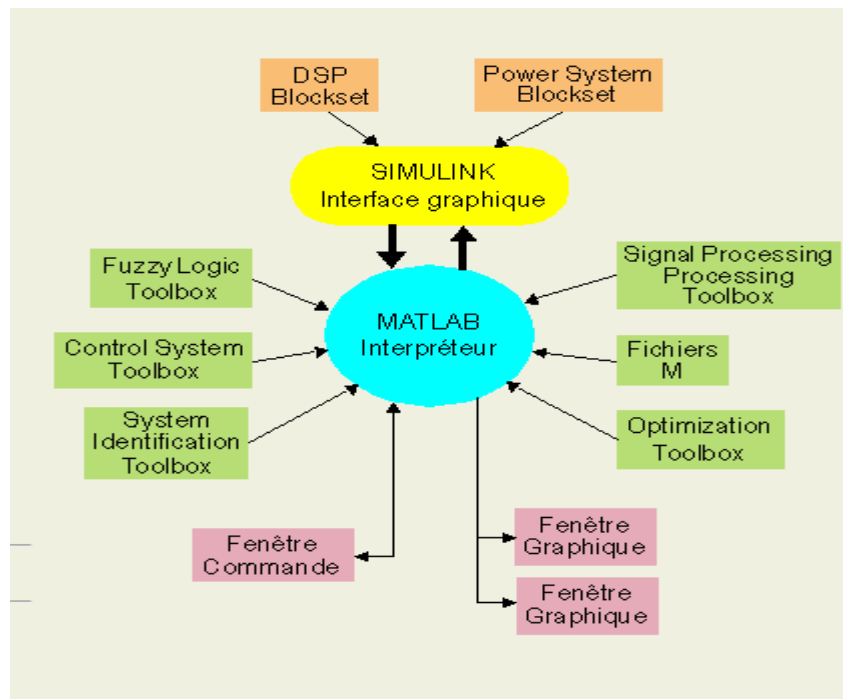
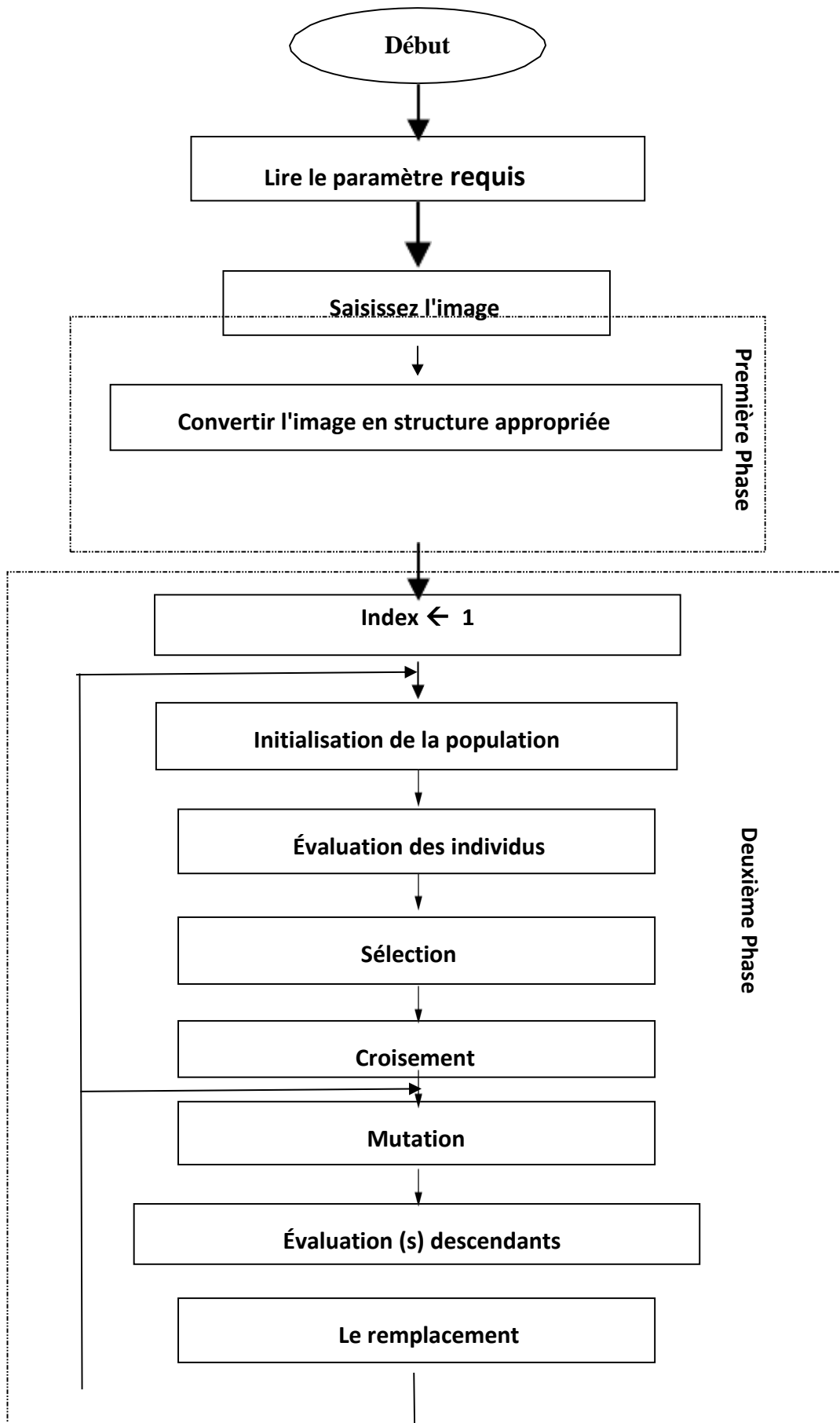


Figure III.1: Schéma programme en langage MATLAB

III.3. Organigramme de l'algorithme (LBG)

Dans cet algorithme, l'image est divisée en un ensemble de vecteurs La série est placée dans un répertoire, puis le montant de la série est choisi parmi Ce guide est répété jusqu'à ce que nous atteignons l'ensemble final de ces Vecteurs de série, et l'algorithme dépend des étapes suivantes :



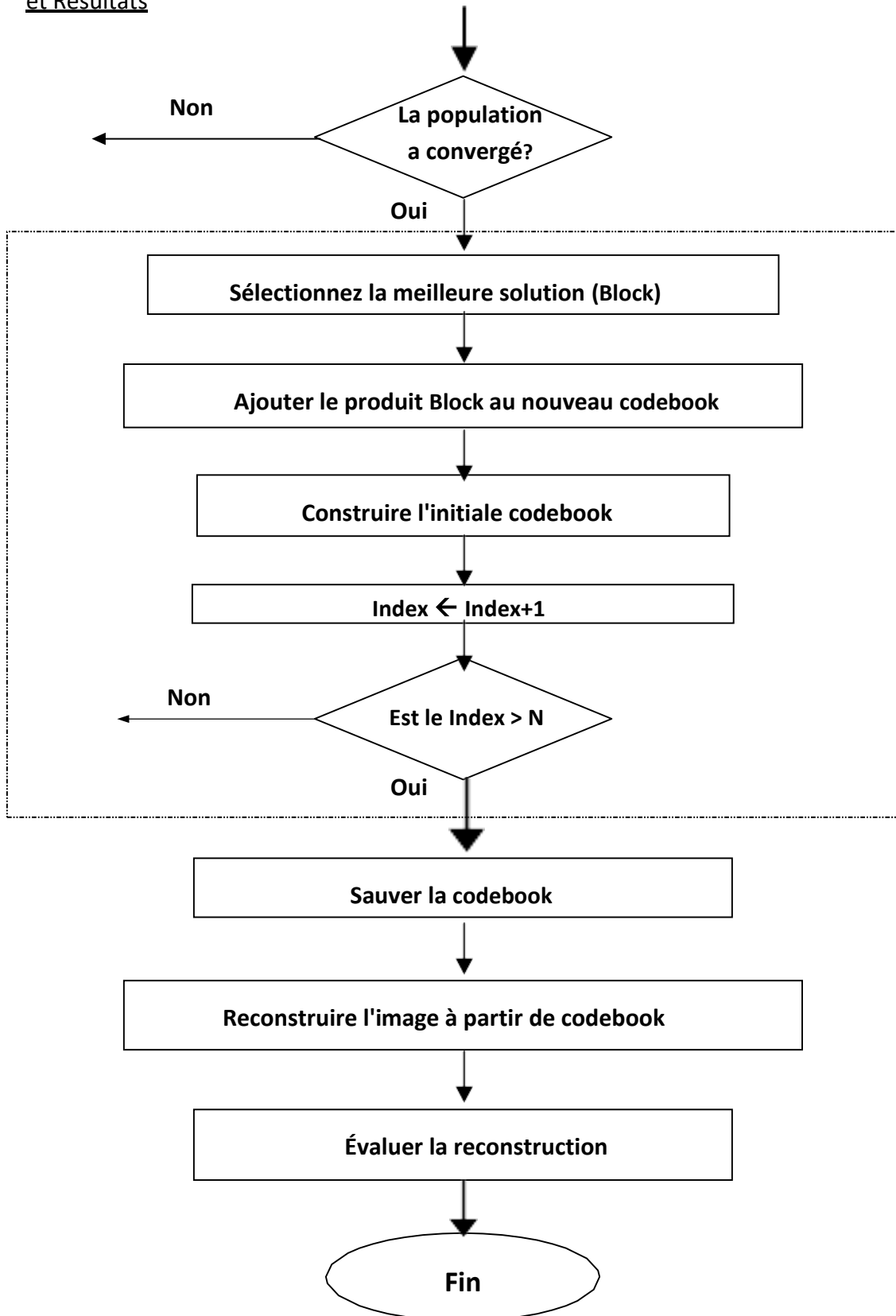


Figure III.2: Organigramme algorithm LBG

Etape 1: Lire les informations requises : Il faut d'abord lire les informations importantes utilisées par le système proposé (ordinateur) et le fichier image à compresser.

Etape 2: Entrée d'image : cette étape consiste à lire les données d'un fichier image BMP où nous utilisons 256 niveaux de gris et une image couleur (512 x 512) pixels.

Etape 3: Convertissez l'image en une structure appropriée : nous diviserons les données d'image en blocs de pixels, où la taille du bloc est importante et affecte la qualité de l'image et le taux de compression. Une taille de bloc plus grande augmentera le taux de compression et diminuera la qualité de l'image (et vice versa). Nous avons trouvé un équilibre entre la qualité et le taux de compression en prenant des blocs d'image de (4 * 4) pixels sur lesquels vous travaillez LBG.

Etape 4: Construire le livre de codes initial : dans cette étape, nous divisons le total des blocs d'images en N unités avec quelques modifications après la classification ; Chaque unité se compose de plusieurs autocollants appartenant à cette unité. N mots de chiffrement sont utilisés pour représenter ces N unités car ce N mot de code est choisi aléatoirement parmi les blocs d'image. Lorsque la taille de l'image est spécifiée, nous exigeons que chaque bloc d'image 4*4 indique uniquement son propre mot de passe 8 bits. L'index de référence de 8 bits permet 256 blocs représentatifs dans le livre de codes. Ce livre de code élémentaire contenant 256 mots de code sélectionnés au hasard à partir de l'image est utilisé pour classer les blocs d'image

Etape 5: Définissez l'index sur 1, où l'index est un compteur du nombre d'unités dans le codebook.

Etape 6: Initialisation de la population : LBG travaillera sur chaque unité dans codebook pour produire le meilleur bloc représentatif qui représente tous les blocs de l'unité. La première étape de LBG consiste à initialiser une communauté de blocs sélectionnés au hasard appartenant au module sur lequel LBG est en cours d'exécution. Le nombre d'individus plus le mot de code dans chaque individu de la communauté se compose de, différence de pixel et bloc d'image (4 * 4 pixels) et chaque gène représente la valeur réelle d'un pixel.

Etape 7: Évaluation des individus : Chaque individu est évalué selon la fonction du bloc le plus représentatif de son unité. Pour chaque individu, nous avons besoin d'une matrice appelée différence de pixel qui contient les différences cumulées du pixel opposé de tous les blocs de la même unité à utiliser dans l'algorithme de calcul de la validité.

Etape 8: le choix: Dans cet article, nous étudions trois méthodes de sélection telles que:

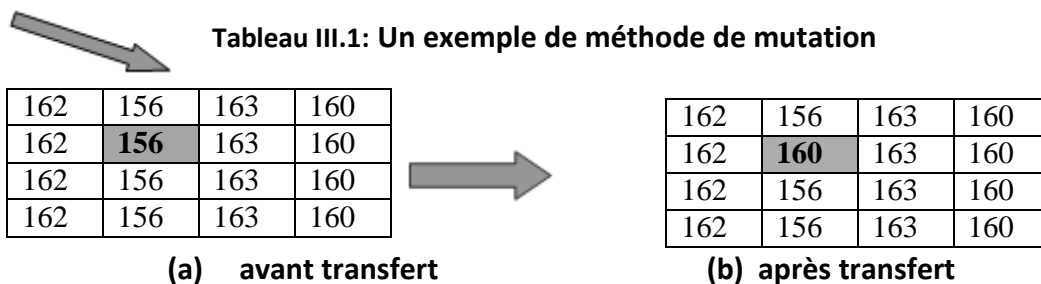
- **Meilleure sélection individuelle (BIS)**
- **Sélection binaire (BIS) :**
- **Sélection hybride (HS):**

Etape 9: Intersection : Définit une propriété de . L'intersection est un processus d'échange. Dans cette étude, nous avons utilisé huit méthodes croisées :

- **1 point de croisement (1x)**
- **intersection de deux points (2x)**
- **croisement unifié (UX)**
- **1 réglage du point de croisement (M1x)**
- **par compte**
- **mélange de churros**
- **croisement de champ**
- **Croisement basé sur la tendance**

Etape 10: Mutation : survient après la procédure de croisement. La mutation change de manière aléatoire. Cela se produit à un certain rythme et est généralement très rare. Le but de la mutagenèse est d'éviter de tomber dans le problème. Dans cette étude, nous utilisons deux des méthodes de mutagenèse proposées :

Mutation 1: Cette méthode traite $n * n$ pixels. Un (pixel) est choisi aléatoirement dans ce (bloc) puis remplacé par les valeurs moyennes des pixels voisins avec sa valeur, comme dans le tableau suivant :



Mutation 2 : Dans cette méthode, lorsqu'une mutation se produit qui est déterminée par la probabilité pertinente, un pixel cible est sélectionné au hasard, puis les valeurs de bloc sont triées. La mutation fonctionne en remplaçant la valeur cible sélectionnée par la valeur moyenne des deux valeurs médianes dans le bloc après tri comme dans le tableau ci-dessous :

III.4.Résultats des simulations

III.4.1 code

```
main.m

load parameter;

global para

x=imread('rice.tif');

original=x;

% v=rand(256,16);

%imwrite(uint8(v),'codebook.bmp');

[v]=trainlvq(x,0);

compressed=v;

%imwrite(uint8(vp),'codebook.bmp');

[y]=testlvq1(x);

[cratio,bpp] = wcompress('c','woodstatue.jpg','woodstatue.wtc',
...
    'aswdr','cc','klt','maxloop',11,'plotpar','step');

wcompress('u','woodstatue.wtc','step');

load mask;

X = imread('wpeppers.jpg');

[cratio,bpp] =
wcompress('c',X,'wpeppers.wtc','spiht','maxloop',12)

subplot(1,2,1)

image(X)

title('Original image')

axis square

subplot(1,2,2)

image(Xc)
```

```
title('Compressed image')  
axis square  
D = abs(double(X)-double(Xc)).^2;  
mse = sum(D(:))/numel(X)  
psnr = 10*log10(255*255/mse)
```

III.4.2 Présentation de l'interface graphique

L'image d'entrée à compresser, définie comme un tableau 2D contenant une image indexée, ou un tableau uint8 3D contenant une image réelle. La taille des lignes et des colonnes de l'image doit être une puissance de deux

```
main.m  
  
load parameter;  
  
global para  
x=imread('rice.tif');  
  
original=x;
```

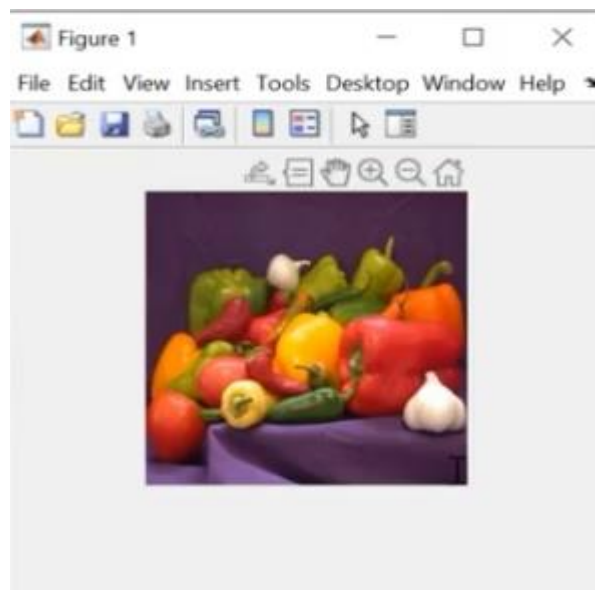


Figure III.3: Voir l'image au programme MATLAB

Méthode de compression, définie comme un vecteur de caractères ou une chaîne scalaire. Les méthodes de compression correctes sont divisées en deux catégories. Méthodes de signification transactionnelle progressive (PCSM).

Tableau III.3: Méthodes de signification transactionnelle progressive (PCSM)

compmethd	Compression method name
'ezw'	Embedded Zerotree Wavelet
'spiht'	Set Partitioning In Hierarchical Trees
'stw'	Spatial-orientation Tree Wavelet
'wdr'	Wavelet Difference Reduction
'aswdr'	Adaptively Scanned Wavelet Difference Reduction

Méthodes de seuil transactionnel (CTM) :

Tableau III.4: Méthodes de seuil transactionnel (CTM)

compmethd	Compression method name
'lv1_mmc'	Subband thresholding of coefficients and Huffman encoding
'gb1_mmc_f'	Global thresholding of coefficients and fixed encoding
'gb1_mmc_h'	Global thresholding of coefficients and Huffman encoding

```

% v=rand(256,16);

%imwrite(uint8(v),'codebook.bmp');

[v]=trainlvq(x,0);

compressed=v;

%imwrite(uint8(vp),'codebook.bmp');

[y]=testlvq1(x);

```

Chapitre III Compression des images par la Algorithmes LBG Simulation et Résultats

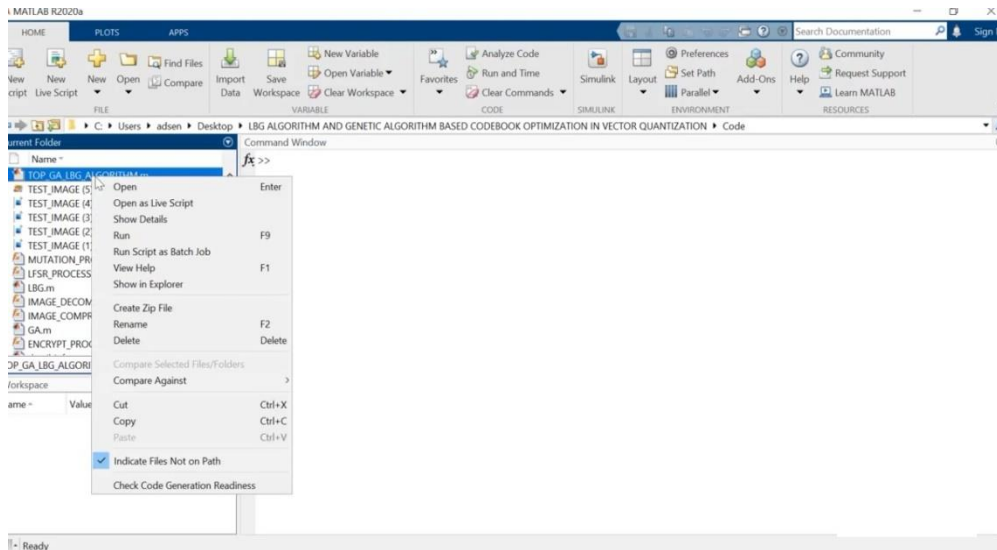


Figure III.4: extrait Méthodes (PCSM) et (CTM) au programme MATLAB

Cet exemple montre comment compresser une image jpeg à l'aide de la méthode de compression par différence d'ondelettes à balayage adaptatif ("aswdr"). La couleur de transformation ("cc") utilise la transformation de Karhunen-Loeve ('kit'). Le nombre maximum de boucles ('maxloop') est défini sur 11 et le type de graphique ('plotpar') est défini pour parcourir la compression. Affichez le taux de compression (cratio) et le taux de bits par pixel (bpp), qui indiquent la qualité de la compression.

III.4.3. Erreur Quadratique Moyenne (EQM) ou Mean Square Error (MSE)

Elle représente l'erreur quadratique moyenne entre l'image compressée et l'image originale. Cette mesure va permettre d'évaluer l'influence des dégradations de la compression sur l'image compressée. Elle est définie comme suit :

$$MSE = \frac{1}{n_x n_y n_\lambda} \sum_{x,y,\lambda} (I(x,y,\lambda) - \tilde{I}(x,y,\lambda))^2$$

III.4.4. Rapport Signal sur Bruit (RSB) ou Signal to Noise Ratio (SNR)

C'est le rapport entre la puissance de l'erreur avec la puissance du signal original (PS).

$$SNR_{(db)} = 10 \log_{10} \frac{PS}{MSE}$$

```
[cratio,bpp] =  
wcompress('c','woodstatue.jpg','woodstatue.wtc', ...  
          'aswdr','cc','klt','maxloop',11,'plotpar','step');  
cratio  
bpp
```

cratio =

3.0792

bpp =

0.7390

```
wcompress('u','woodstatue.wtc','step');
```



Figure III.5: Voir l'image compressée au programme MATLAB

Cet exemple montre comment compresser une image truecolor à l'aide d'un partitionnement défini dans des arbres hiérarchiques - une méthode de compression 3D ('spiht_3D'). Charger, compresser et stocker l'image dans un fichier. Dessinez les images originales et compressées. Affiche le taux de

Chapitre III Compression des images par la Algorithmes LBG Simulation et Résultats

compression ('cratio') et les bits par pixel ('bpp'), qui indiquent la qualité de la compression.

```
load mask;  
X = imread('wpeppers.jpg');  
[cratio,bpp] =  
wcompress('c',X,'wpeppers.wtc','spiht','maxloop',  
12)
```

cratio = 1.6549

bpp = 0.3972

Visualisez les images originales et compressées.

```
subplot(1,2,1)  
image(X)  
title('Original image')  
axis square  
subplot(1,2,2)  
image(Xc)  
title('Compressed image')  
axis square
```

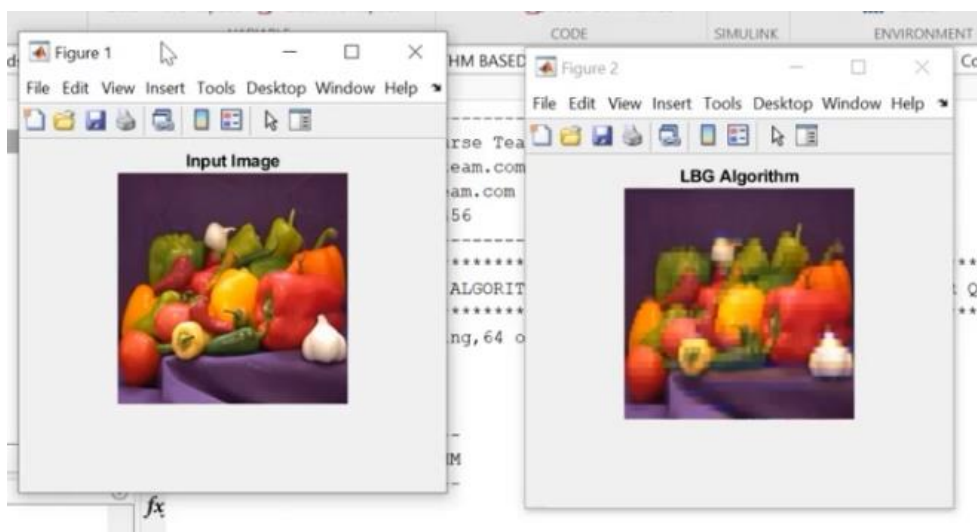


Figure III.6: Voir l'image originale et l'image compressée au programme MATLAB

Chapitre III Compression des images par la Algorithmes LBG Simulation et Résultats

Calculez l'erreur quadratique moyenne (MSE) et les valeurs d'erreur du rapport signal sur bruit (PSNR). Vous pouvez utiliser ces deux échelles pour déterminer l'erreur entre deux images. Le PSNR est exprimé en decibels.

```
D = abs(double(X)-double(Xc)).^2;  
mse = sum(D(:))/numel(X)
```

```
mse = 257.63
```

```
psnr = 10*log10(255*255/mse)
```

```
psnr = 49.042
```



```
Command Window  
EXECUTING LBG ALGORITHM  
-----  
EXECUTING GENETIC ALGORITHM  
-----  
MSE_RESULT =  
1x2 table  
    LBG_ALGORITHM_MSE    GENETIC_ALGORITHM_MSE  
-----  
                257.63                239.63  
-----  
PSNR_RESULT =  
1x2 table  
    LBG_ALGORITHM_PSNR    GENETIC_ALGORITHM_PSNR  
-----  
                48.042                48.671  
-----  
fx >> |
```

Figure III.7: Voir (MSE) et (PSNR) au programme MATLAB.

III.5 Testes et Résultats de la compression

L' algorithme mentionnée a été mise en oeuvre sous Matlab R2008b et les performances ont été évaluées pour différents facteur de qualité : taux de compression(T_c), l'erreur quadratique moyenne (MSE),le PSNR,la taille de l'image décompressée et sa qualité . Nous avons obtenu les résultats suivants pour différentes valeurs du facteur de qualité (FQ).

❖ Avec FQ=1



Image original 512x512



I mage compressée



I mage décompressée

Figure III.8 Voir deux l'image originale et compressée FQ= 1

Tableau III.5 Résultats de la compression avec FQ=1

Taille de l'image original (bit)	Taille de l'image Compressée(bit)	Taux de compression	PSNR (db)	MSE
786432	675469	1.16	43.77	2.7

❖ Avec FQ=3



Image original 512x512



I mage compressée



I mage décompressée

Figure III.9 Voir deux l'image originale et compressée FQ=3

Tableau III.6 Résultats de la compression avec FQ=3

Taille de l'image original (bit)	Taille de l'image Compressée(bit)	Taux de compression	PSNR (db)	MSE
786432	340588	2.30	39.26	7.70

❖ Avec FQ=10



Image original 512x512



I mage compressée



I mage décompressée

Figure III.10 Voir deux l'image originale et compressée FQ=10

Tableau III.7 Résultats de la compression avec FQ=10

Taille de l'image original (bit)	Taille de l'image Compressée(bit)	Taux de compression	PSNR (db)	MSE
786432	252159	3.11	34.46	23.27

❖ Avec FQ=20



Image original 512x512



I mage compressée



I mage décompressée

Tableau III.8 Résultats de la compression avec FQ=20

Taille de l'image original (bit)	Taille de l'image Compressée(bit)	Taux de compression	PSNR (db)	MSE
786432	133916	5.87	31.69	43.99

❖ Avec FQ=50



Image original 512x512



I mage compressée



I mage décompressée

Figure III.12 Voir deux l'image originale et compressée FQ=50

Tableau III.9 Résultats de la compression avec FQ=50

Taille de l'image original (bit)	Taille de l'image Compressée(bit)	Taux de compression	PSNR (db)	MSE
786432	106364	7.39	27.78	108.20

❖ Avec FQ=80



Image original 512x512



I mage compressée



I mage décompressée

Figure III.13 Voir deux l'image originale et compressée FQ=80

Tableau III.10 Résultats de la compression avec FQ=80

Taille de l'image original (bit)	Taille de l'image Compressée(bit)	Taux de compression	PSNR (db)	MSE
786432	99672	7.89	25.99	163.35

On résume toute les résultats de compression pour les déférentes valeurs

FQ dans le tableau suivant :

Tableau III.11 Résultats de la compression avec FQ= (1 jusqu'à 80)

FQ	Taille de l'image Compressée(bit)	T _C	PSNR (db)	MSE	G _C (%)
1	675469	1.16	43.77	2.72	14.10
3	340588	2.30	39.26	7.70	56.69
5	252159	3.11	37.16	12.48	67.93
10	177041	4.44	34.46	23.27	77.48
20	133916	5.87	31.69	43.99	82.97
50	106364	7.39	27.78	108.20	86.47
80	99672	7.89	25.99	163.35	87.32

- ✓ D'après les résultats visuels de la compression, on remarque que dans le cas de (FQ=10) l'image a été compressée avec taux de compression T_C = 4.44 ce qui est acceptable. L'image n'a pas perdu de sa fidélité apparente, cela étant seulement visuellement, apparent, parce que l'erreur quadratique moyenne n'est pas nulle et égale à EQM = 23.27 cela veut dire qu'il y'a une erreur d'une différence moyenne environs 23.3 entre les éléments des deux matrices (matrice originale /matrice décompressée).
- ✓ Le rapport crête signal sur bruit PSNR=34.46 dB est acceptable, ce qui montre qu'on a fait une compression efficace en matière de taille/qualité.
- ✓ on remarque que plus on choisit un facteur de qualité grand, plus le taux de compression est meilleur du fait qu'on augmente le seuil de la table Q mais MSE augmente aussi ce qui va influencer sur l'efficacité de la compression constatée visuellement et aussi au vue du rapport PSNR médiocre du fait qu'on a perdus beaucoup de données pendant l'étape de

quantification et cela est devenue très visible, ce qui rend l'image exploitable à certaines limites.

III.6 Conclusion

Dans ce chapitre, nous avons décrit un algorithme de (compression / décompression) des images basé sur la transformation en Cosinus séparé pour changer la base de la matrice d'image de la base spatiale à la base de fréquence, et en même temps pour séparer les fréquences inférieures, les informations d'image dans le coin supérieur gauche de chaque bloc 8×8 . Ensuite, la quantification a été effectuée jusqu'à la taille de fichier effective de l'image était beaucoup plus petite. 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.

Conclusion générale

Conclusion générale

La compression permet de réduire la taille de images nécessaire à leur archivage et de diminuer le temps nécessaire à leur transmission à distance via les réseaux informatiques.

La compression dépend de l'algorithme LBG. Nous avons étudié et comparé les performances de compression sans perte sur des images en niveaux de gris et des images de fruits. C'est pourquoi nous avons utilisé . Les métriques utilisées pour déterminer la qualité de la pression sont le taux de compression, TC., et PSNR.

Actuellement, la méthode de compression d'image est parmi les plus utilisées car elle permet d'atteindre des taux de compression très élevés sans que l'œil humain ne détecte les changements dans l'image. De plus, de nombreuses applications permettent de choisir la qualité de l'image compressée grâce à l'utilisation de Codebook. La réputation et, par conséquent, le nombre d'utilisateurs de l'algorithme LBG de compression dépend de divers facteurs : le rapport taille/qualité, la vitesse de compression et de décompression. Il est donc intéressant de comparer les différentes méthodes, pour pouvoir choisir et utiliser la plus adaptée à nos besoins.

Bibliographie

- [1] KADDOUR Chakib, « Généralités sur le traitement d'images », 1999
- [2] R. HAROUN- S. A. BOUHRICHE, « Séparation des chromosomes se touchant ou se chevauchant à partir de Meta phases », diplôme d'ingénieur d'état en Electronique,
ENP- Promotion 2001-2002
- [3] <http://www.commentcamarche.net/video>
- [4] Gregory A Baxes "digital image processing",prentice _hall,1984
- [5] M.Colinet« Le traitement des images », Namur-Faculté universitaire Notre-Dame DE LA PAIX Mai 2004.
- [6] BOUCETA, AEtude de l'effet des transformées des décorrélation en compression des images couleur RGB (Doctoral dissertation, Batna, Université El Hadj Lakhdar.Faculté des sciences de l'ingénieur) (2010).
- [9] C.BOUDRY. « Qu'est –ce qu'une image numérique ».
- [11] D'aviné, compression d'image par Fractales Basé sur la triangulation de Delaunay,thèse Institut National polytechnique de Grenoble, Décembre 1995.
- [7] C. Diltsm,Numérisés vos collection, RCIP OCTOBRE 2001.<http://www.ching.gc.ca/Français/contenu.Numerique/Guide.Gestionnaire/>
- [8] Jean FRUIT « outils et méthode pour le traitement des images par ordinateur » URL : [http:// etudiant.mev.fr/fruite/trait. Image./an-image.html](http://etudiant.mev.fr/fruite/trait.Image./an-image.html) URFIST de paris2002.2010.
- [9] Rafael G. Gonzalez and Richard E. Woods, “Digital Image Processing”, Addison - Wesley, 1993.
- [10]Rafael G. Gonzalez Paul Wintz, "Digital Image Processing", Addisonesley, 1977.
- [11] Gregory A.Baxes, “Digital Image Processing”, John Wiley and Sons. Inc., 1994
- [12] Louis J. Galbiati.Jr, "Machine Vision and Digital Image Processing Fundamentals" Don Pearson, "Image Processing", McGraw, 1991,

Bibliographie

- [13] Thomas M. Lilleasand and Ralph W. Kiefer, "Remote Sensing and Image Interpretation", John Wiley and Sons, 1985.
- [14], 1990, 7 المجلد 6 العدد الكويتية العلوم مجلة بايلور p62-p70
- [15]. عبد الفتاح رياض، "الإضاءة والفيلم في التصوير الضوئي"، مكتبة لأنجلو المصرية.
- [16] Craig A. Lindley, "Practical Image Processing", John Wiley and Sons, 1991.
- [17] Gregory A. Baxes, "Digital Image Processing", John Wiley and Sons. Inc., 1994.
- [1] Kekre .H. B. and Tanuja K. Sarode " New Clustering Algorithm for Vector Quantization using Rotation of Error Vector" IJCSIS International Journal of Computer Science and Information Security, Vol. 7, No. 3, 2010 ,India
- [18] B. and Tanuja K. Sarode " New Clustering Algorithm for Vector Quantization using Rotation of Error Vector" IJCSIS International Journal of Computer Science and Information Security, Vol. 7, No. 3, 2010 ,India
- [19] Aza-aoba 05 , Aramaki , Aoba-ku, Sendai , Gray of 2001 , Robeit.M.Gray Information System Laboratory 980-8579, Japan . 2006
- [20] Zhibin Pan¹, koji kotani², and Tadahiro Ohmi¹ , , A Fast Arch Method For Vector Quantization Using Pixel-Merging Sum Pyramid In Recursive Way, Department of Electronic Engineering , Graduate School of Engineering , Tohoku University , Japan , Aza-aoba 05 , Aramaki , Aoba-ku, Sendai , 980-8579, Japan . 2005
- [21] Gray of 2001 , Robeit.M.Gray Information System Laboratory.
- [22] «<https://fr.wikipedia.org/w/index.php?=&compression-dimage&olid=162657565> »
- [23] Mohammed Ben Abdellah, « Outils de compression et de crypto compression : Application aux images fixes et vidéo », Thèse de Doctorat, Université Mohammed V- agdal, Rabat-Maroc, 2007
- [24] Chassery J -M. M. Melkemi, « Diagramme de voronoi applique à la segmentation d'images et à la détection d'événements en imagerie multi-sources », Traitement du signal, vol.8, N°03, pp.155-164, 1991
- [25] J. Fages « JPEG2000 – Principes, implémentation et évaluation », Mémoire d'ingénieur, Université de Houari Boumediene, 2002
- [26]. A. A. Abdelwahab and N. S. Muharram, "A Fast Codebook Design Algorithm Based on a Fuzzy Clustering Methodology," International Journal of Image and Graphics, vol. 7, No. 2, pp. 291-302, 2007.

Bibliographie

- [27]. C. Garcia and G. Tziritas, "Face Detection using Quantized Skin Color Regions Merging and Wavelet Packet Analysis," *IEEE Trans. Multimedia*, vol. 1, No. 3, pp. 264–277, Sep. 1999
- [28]. H. B. Kekre, T. K. Sarode and B. Raul, "Color Image Segmentation using Kekre's Algorithm for Vector Quantization," *International Journal of Computer Science (IJCS)*, vol. 3, No. 4, pp. 287-292, Fall 2008.
- [29]. H. B. Kekre, T. K. Sarode and B. Raul, "Color Image Segmentation using Vector Quantization Techniques Based on Energy Ordering Concept," *International Journal of Computing Science and Communication Technologies (IJCSCT)*, vol. 1, Issue 2, January 2009.
- [30]. H. B. Kekre, V. A. Bharadi, S. Ghosalkar and R. A. Vora, "Blood Vessel Structure Segmentation from Retinal Scan Image using Kekre's Fast Codebook Generation Algorithm," *Proceedings of the International Conference & Workshop on Emerging Trends in Technology, ACM, ICWET'2011, NY, USA*, pp. 10-14, 2011.
- [31]. H. B. Kekre and T. K. Sarode, "Speech Data Compression using Vector Quantization," *WASET, International Journal of Computer and Information Science and Engineering, (IJECSE)*, vol. 2, Number 4, pp. 251-254, 2008.
- [32]. H. B. Kekre, T. K. Sarode and S. D. Thepade, "Image Retrieval using Color-Texture Features from DCT on VQ Codevectors obtained by Kekre's Fast Codebook Generation," *ICGST-International Journal on Graphics, Vision and Image Processing (GVIP)*, vol. 9, Issue 5, pp. 1-8, September 2009.
- [33]. H. B. Kekre, T. K. Sarode and S. Gharge, "Detection and Demarcation of Tumor using Vector Quantization in MRI images," *International Journal of Engineering Science and Technology*, vol.1, Number 2, pp. 59-66, 2009.
- [34]. H. B. Kekre, T. K. Sarode and S. Gharge, "Kekre's Fast Codebook Generation algorithm for tumor detection in mammography images," **Proceedings of the**

Bibliographie

International Conference and Workshop on Emerging Trends in Technology, ICWET'10, ACM, pp. 743-749, NY, USA, 2010.

[35]. C. M. Huang and R. W. Harris, "A Comparison of Several Vector Quantization Codebook Generation Approaches," IEEE Trans. On Image Processing, vol. 2, No. 1, pp. 108-112, 1993

[36]. Hsieh.C.H. , Lu.P.C. and Chung. J.C, "Fast Codebook Generation Algorithm for Vector Quantization of Images", Pattern Recognition Letter, 12, pp. 605-609. 1991.

[37] Katzenbeisser, S. and Petitcolas, F.A.P., " Information Hiding Techniques for Steganography and Digital Watermarking." Artech House,Inc.,Boston,London,2000.

[38] Y. Linde, A. Buzo, R.M. Gray, " An algorithm for Vector quantiser design", IEEE Transaction on Communications, Vol.28, pp.84-95, 1980. [3] Guorong Xuan et.al,DzDistortionless Data Hiding Based On IWTdz, Electronics Letters, vol.,8,No.'и,pp.ТЭИЭ- 1648,.2002.

[39] Shejul ,A. A.,Kulkarni, U.L., DzA Secure Skin Tone based Steganography (SSTS) using Wavelet Transformdz,International Journal of Computer Theory and Engineering, Vol.3,No.1,pp.16-22,2011.

[40] Masud,Karim S.M., Rahman, M.S.,Hossain, M.I., DzA New Approach for LSB Based Image Steganography using secret key.dzТИth International conference on Computer and Information Technology, IEEE Conference Publications, pp 286-291,2011

Résumé

Dans cet article, un algorithme de quantification d'image à deux phases basé sur l'algorithme génétique en régime permanent (LBG) pour la compression d'image est proposé. L'objectif de la première phase est de créer le livre de codes initial utilisé pour catégoriser les blocs d'images en fonction de certaines mesures de distorsion, tandis que le LBG produira le meilleur bloc représentatif qui représente tous les blocs de chaque unité du livre de codes dans la deuxième phase. . Les résultats de la simulation expliquent que le LBG proposé présente un bon taux de compression avec une image reconstruite de haute qualité. Mots clés : compression d'image, quantification vectorielle, algorithmes génétiques.

Mot clé Compression d'image, LBG, Codebook, niveaux gris, Matlab

Hammoudi mabrouk Félicitations. Département Electroniques, Faculté des Sciences Technologiques, Université Amar Teleji de Laghouat, Algérie

Abstract:

In this paper, A steady – state Genetic Algorithm (LBG) based two phase image quantization algorithm for image compression is proposed. The aim of the first phase is create the initial codebook that used to categorize the image blocks by a according to some distortion measures, while the LBG will produce the best representative Block that represents the all blocks in each unit within the codebook in the second phase. The simulation results explain that the proposed LBG exhibits a good compression ratio with high quality reconstructed image. Keywords: image compression, vector quantization, Genetic Algorithms.

Key Word Image Compression, LBG, Codebook, Grayscale, Matlab

Hammoudi mabrouk. Electronics Department, Faculty of Technological Sciences, Amar Teleji University of Laghouat, Algeria

الخلاصة

أُقترح في هذا البحث، طريقة تستخدم الخوارزمية (LBG)، لضغط الصور. باستخدام المكمم الاتجاهي بمرحلتين. إن الهدف من المرحلة الأولى هي لخلق codebook ابتدائي يستخدم لتصنيف بلوكات الصورة وفقا لمقياس خطأ معين، بينما تستخدم خوارزمية LBG للحصول على أفضل تمثيل للبلوك الذي يمثل كل البلوكات الموجودة في كل وحدة في داخل ال-codebook في المرحلة الثانية، توضح النتائج التي تم الحصول عليها باستخدام الطريقة المقترحة بأن نسبة الضغط جيدة مع جودة عالية للصورة المسترجعة.

الملحقات الرئيسية: ضغط الصور ، LBG ، Codebook ، niveaux gris ، ماتلاب

حمودي مبروك . قسم علوم الالكترونيك ، كلية العلوم التكنولوجية، جامعة عمار تليجي بالاغواط الجزائر