

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE DE LA RECHERCHE SCIENTIFIQUE



**Université Mohammed Seddik
Ben Yahia –Jijel–**

**Faculté des Sciences et de la Technologie
Département d'Electronique**



**Projet de fin d'études pour l'obtention du diplôme de
Master en Télécommunications**

Option

Systèmes des Télécommunications

Thème

**Compression d'images sans perte
par la méthode du codage prédictif**

Présenté par :

M^r Brahim ABDI

Encadré par :

Dr. Tahar BRAHIMI

Année universitaire : 2020-2021

Remerciements

*Tout d'abord nous remercions ALLAH, le tout puissant de nous avoir
donné la*

force et le courage pour accomplir ce modeste travail de recherche.

*Nous remercions vivement notre encadreur M^r T. BRAHIMI de nous
avoir confié*

*ce travail d'abord et pour son soutien, ses conseils précieux, ses
encouragements et*

sa grande patience durant toute la période de ce travail.

*Nous remercions les membres du jury, qui nous ont fait l'honneur de
participer*

au jugement de ce travail.

*Nous souhaitent également remercier tous nos enseignants d'Electronique
pour*

*tous le savoir qu'ils nous ont donné durant toute ces années d'étude, plus
précisément notre agréable chef de département Mr T. BENKEDIDAH,*

sans oublier sa secrétaire Mlle B. BADIA pour sa gentillesse.

*Un remerciement très chaleureux à nos parents pour leurs soutiens et leurs
conseils tout au long de nos études ainsi que tous les membres de la
famille, nos*

*amis en particulier K. YAKOUB et tous ceux, qui de près ou de loin, nous
ont apportés leur contribution, pour accomplir ce travail.*

Dédicace

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes chères sœurs

Pour leurs encouragements permanents, et leur soutien moral,

A mes chers frères,

Pour leur appui et leur encouragement,

A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible,

Merci d'être toujours là pour moi.

Brahim Abdi

Sommaire

Sommaire	III
Liste des figures	VI
Liste des tableaux	VII
Liste des abréviations	VIII
Introduction générale	1

Chapitre I

Généralités et Notions de Base sur l'image

I.1. Introduction.....	4
I.2. Définition d'une image	4
I.3. Caractéristiques d'une image.....	4
I.3.1. Image numérique.....	4
I.3.2. Définition d'un pixel	5
I.3.3. La résolution.....	5
I.3.4. Dimension	5
I.3.5. Bruit.....	6
I.3.6. La luminance	6
I.4. Représentation d'une l'image	6
I.5. Types d'images	7
I.5.1. Images binaires (en noir et blanc)	7
I.5.2. Images en niveaux de gris (Monochromes)	8
I.5.3. Images en couleur (Polychromes)	8
I.6. Les différents formats d'image	9
I.7. Définition de la compression	10
I.7.1. Différence entre la compression avec perte et la compression sans perte.....	11
I.8. Définition des critères utilisés en compression.....	11
I.8.1. Taux de compression.....	11
I.8.2. Le débit.....	12
I.8.3. Critères de qualités.....	12
I.9. Quantité d'information et l'entropie.....	12

I.10. La compression JPEG	13
I.10.1. Principe de compression du JPEG	13
I.10.2. Utilité et caractéristiques de la compression JPEG	13
I.11. Conclusion	13

Chapitre II

Techniques de codage sans perte

II.1. Introduction	15
II.2. Définition du codage.....	15
II.3. Les différentes méthodes de compression sans perte	15
II.3.1. Le codage de Shannon-Fano	15
II.3.2. Le codage de Huffman.....	17
II.3.3. Le codage Arithmétique.....	18
II.3.3.1. Procédure du codage Arithmétique	19
II.4. Codage à base de dictionnaire	19
II.4.1. RLE (Run-length-encoding)	20
II.4.2. LZW (Lempel Ziv Welch)	20
II.5. Le codage prédictif	24
II.5.1. Codage prédictif sans perte	24
II.6. Histogramme (Cameraman)	26
II.7. Application à la compression d'images	26
II.7.1. Algorithme du codage de Huffman (CH)	27
II.7.2. Algorithme du codage Arithmétique (CA)	27
II.8. Algorithmes basés sur une étape de codage prédictif.....	28
II.8.1. Algorithme du Codage de Huffman associé au codage prédictif (CHP)	28
II.8.2. Algorithme du codage Arithmétique associé au codage prédictif (CAP).....	29
II.9. Conclusion	29

Chapitre III

Simulations et résultats

III.1. Introduction	31
III.2. Les données de test et leurs caractéristiques	31
III.3. Machine PC	31
III.4. Evaluation de la performance en termes de débit (en Bpp)	31
III.4.1. Evaluation de la performance des techniques de codage: CH (codage de Huffman) et CHP (codage de Huffman associé au codage prédictif)	31
III.4.2 Evaluation de la performance des techniques de codage: CA (codage arithmétique) et CAP (codage arithmétique associé au codage prédictif)	34
III.3 Evaluation de la performance en termes de complexité de calcul (temps de codage et décodage)	37
III.6. Comparaison de la performance des techniques de codage utilisant le codage prédictif, les techniques CHP et CAP en termes de débit (en Bpp).....	40
III.7 Conclusion.....	43
Conclusion générale	44
Références Bibliographiques	45

Liste des figures

Figure I. 1: Représentation d'image numérique.....	4
Figure I. 2: Groupe de pixels formant la lettre A.....	5
Figure I. 3: (a) Image sans bruit. (b) Image avec bruit.....	6
Figure I. 4: Image en noire et blanc.	7
Figure I. 5: Image monochrome.....	8
Figure I. 6: Image polychrome (couleur).	9
Figure I. 7: Les étapes de la Compression.....	13
Figure I. 8: Les étapes de la décompression.....	13
Figure II. 1: Algorithme de Shannon-Fano sur l'alphabet.....	16
Figure II. 2: La procédure de codage de Huffman à variance minimale.....	18
Figure II. 3: Un modèle de codage prédictif sans perte : codeur et décodeur.	25
Figure II. 4: Un histogramme d'une image original.....	26
Figure II. 5: Histogramme de l'erreur de prédiction.	26
Figure II. 6: Schéma de compression.....	27
Figure II. 7: Schéma de décompression.....	27
Figure II. 8: Schéma de compression.	28
Figure II. 9: Schéma de décompression.....	28
Figure III. 1: Gain en bits par pixel entre les techniques CH et CHP.	34
Figure III. 2: Gain en bits par pixel entre les techniques CA et CAP.	37
Figure III. 3: (a) Temps de codage (b) Temps de décodage en secondes entre les techniques CH et CHP.....	39
Figure III. 4: (c) Temps de codage (d) Temps de décodage en secondes entre les techniques CH et CHP.....	40
Figure III. 5: Gain en bits par pixel entre les techniques CHP et CAP.	42

Liste des tableaux

Tableau I. 1: Les différents formats d'images.....	10
Tableau II. 1: Fréquences d'apparition des symboles de l'alphabet.....	16
Tableau II. 2: Code de Shannon-Fano pour l'alphabet.....	16
Tableau II. 3: Code de Huffman à variance minimale.	18
Tableau II. 4: Codage LZW.....	22
Tableau II. 5: Décodage LZW.....	23
Tableau III. 1: Tableau comparatif des performances de compression sans perte des techniques de codage: CH et CHP.	32
Tableau III. 2: Tableau comparatif des gains de débits (en Bpp) entre les techniques de codage CH et CHP	34
Tableau III. 3: Tableau comparatif des performances de compression sans perte des techniques de codage: CA et CPA.	35
Tableau III. 4: Tableau comparatif des gains de débits (en Bpp) entre les techniques de codage CA et CAP.	36
Tableau III. 5: Comparaison des performances des deux techniques de codage CH et CHP en termes de temps de codage et décodage (en secondes).	38
Tableau III. 6: Comparaison des performances des deux techniques de codage CA et CAP en termes de temps de codage et décodage (en secondes).	38
Tableau III. 7: Tableau comparatif des performances de compression sans perte des techniques de codage: CPH et CPA.	41
Tableau III. 8: Tableau comparatif des gains de débits (en Bpp) entre les techniques de codage CHP et CAP	42

Liste des abréviations

Bpp	Bits par pixel
RVB	Rouge Vert Bleu
CMJ	Cyan Magenta Jaune
CH	Codage de Huffman
CA	Codage d'Arithmétique
CHP	Codage de Huffman associé au codage Prédicatif
CAP	Codage d'Arithmétique associé au codage Prédicatif
TIFF	Tagged Image File Format
JPEG	Joint Photographic Expert Group
GIF	Graphics Interchange Format
PNG	Portable Network Graphic
DCT	Discret Cosine Transform
MSE	Mean Square Error
SNR	Signal to Noise Ratio
PSNR	Peak Signal to Noise Ratio
SSIM	Structural SIMilarity
RLE	Run-length-encoding
LZW	Lempel-Ziv-Welch

Introduction générale

La compression de données est l'art ou la science de représenter l'information sous une forme compacte. Ces représentations compactes sont créées en identifiant et en utilisant les structures qui existent dans les données. Les données peuvent être des caractères dans un fichier texte, des nombres qui sont des échantillons de signaux de parole ou d'image, ou des séquences de nombres générées par d'autres processus. La compression des données est nécessaire car de plus en plus d'informations que nous générons et utilisons sont sous forme numérique, consistant en des nombres représentés par des octets de données. Dans ce contexte, le nombre d'octets nécessaires pour représenter des données multimédia peut être important [1].

Sur la base des exigences de la reconstruction, les techniques de compression peuvent être distinguées en deux grandes classes: la compression sans perte et la compression avec perte. Il sera question de compression sans perte lorsque l'image (ou le signal) reconstruite est identique à l'image (ou signal) d'entrée. Lorsque l'image reconstruite est différente de l'image d'entrée, on parlera alors de compression avec perte.

Il existe de nombreuses méthodes connues pour la compression de données [2]-[9]. Elles sont basées sur des idées différentes, conviennent à différents types de données et produisent des résultats différents, mais elles reposent toutes sur le même principe, à savoir qu'elles compressent les données en supprimant la redondance des données originales dans le fichier source.

La corrélation entre les échantillons impose une structure aux données qui peut être exploitée pour la compression de différentes manières. Afin de réduire les redondances inter pixels, l'image d'entrée doit être transformée en un format plus efficace.

En raison de la corrélation qui existe entre les échantillons voisins ayant des valeurs similaires, nous pouvons utiliser un codage prédictif qui permet de prédire la valeur de l'échantillon que nous essayons de coder en nous basant sur les valeurs des échantillons précédents avec un certain degré de précision. La différence entre les valeurs réelles et prédites devrait alors, en moyenne, avoir une plage dynamique plus petite que la plage dynamique des valeurs de l'échantillon original. En effet, nous regroupons ou agrégeons à nouveau les valeurs

qui doivent être représentées, ce qui nous permet d'utiliser par la suite un codeur entropique de manière plus efficace [1] [10] [9].

Notre projet de fin d'études a pour but d'évaluer et analyser la performance de compression sans perte de deux techniques de codage de source, à savoir le codage de Huffman et le codage arithmétique, appliquées indépendamment et conjointement avec le codage prédictif. A cette fin, nous exposons les résultats et analysons les performances de quatre variantes nommées: CH (Codage de Huffman), CHP (codage de Huffman associé au codage prédictif), CA (Codage arithmétique) et CAP (codage arithmétique associé au codage prédictif) en mesurant le débit (exprimé en bits par pixel) et en illustrant la complexité de calcul par la mesure du temps de codage/décodage, en utilisant des images en niveaux de gris de taille variable avec des caractéristiques bien différentes.

Pour favoriser l'atteinte de ces objectifs, le mémoire comporte trois chapitres.

Le premier chapitre de ce mémoire sera consacré à l'introduction des notions de base et les principes est concepts attachés aux domaines des images et aussi à l'image elle-même.

Le deuxième chapitre décrit différentes techniques du codage, utilisées couramment en compression sans perte. Il présente également quatre variantes de techniques de compression d'images sans perte par la combinaison de techniques du codage source et du codage prédictif.

Le troisième chapitre présent les résultats obtenus par notre étude portant sur la compression sans perte en s'appuyant sur une évaluation objective.

Le mémoire s'achève par une conclusion générale.

Chapitre I

Généralités et Notions de Base sur l'image

- I.1. Introduction
 - I.2. Définition d'une image
 - I.3. Caractéristiques d'une image
 - I.4. Représentation d'une l'image
 - I.5. Types d'images
 - I.6. Les différents formats d'image
 - I.7. Définition de la compression
 - I.8. Définition des critères utilisés en compression
 - I.9. Quantité d'information et l'entropie
 - I.10. La compression JPEG
 - I.11. Conclusion
-

I.1. Introduction

La compression d'image peut être définie comme l'ensemble des méthodes et techniques opérant sur l'image afin d'extraire les informations les plus pertinentes pour fournir une image plus perceptible à l'œil humain avec une taille réduite. Dans ce chapitre, nous explorons quelques notions de base faisant référence à l'image et ses différents types, afin d'étudier la compression d'image. Ensuite, nous présenterons les critères utilisés en compression pour évaluer la performance d'une méthode de compression.

I.2. Définition d'une image

L'image est une représentation d'une personne ou d'un objet par la peinture, le dessin, la photographie, le film, etc. C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain.

Elle peut être décrite sous la forme d'une fonction $I(x, y)$ 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, ce qui nécessite sa numérisation [11].

I.3. Caractéristiques d'une image

I.3.1. Image numérique

L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs [11].

La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques $f(x, y)$, comme la montre la figure I.1 où : x, y : Les coordonnées cartésiennes d'un point de l'image. $f(x, y)$: Le niveau d'intensité. La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur.

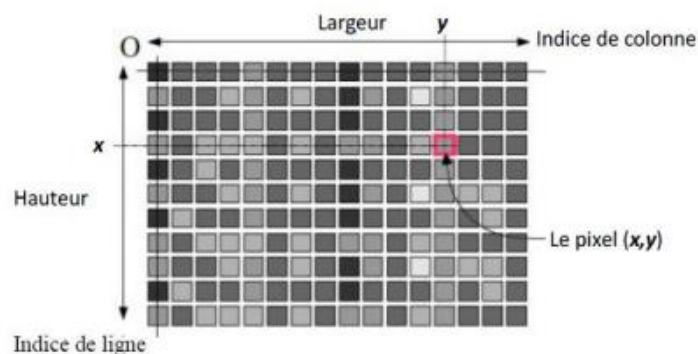


Figure I. 1: Représentation d'image numérique

I.3.2. Définition d'un pixel

Contraction de l'expression anglaise " Picture Elements ": éléments d'image, le pixel est le plus petit point de l'image, c'est une valeur numérique représentative des intensités lumineuses.

Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels sur l'image. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous [12].

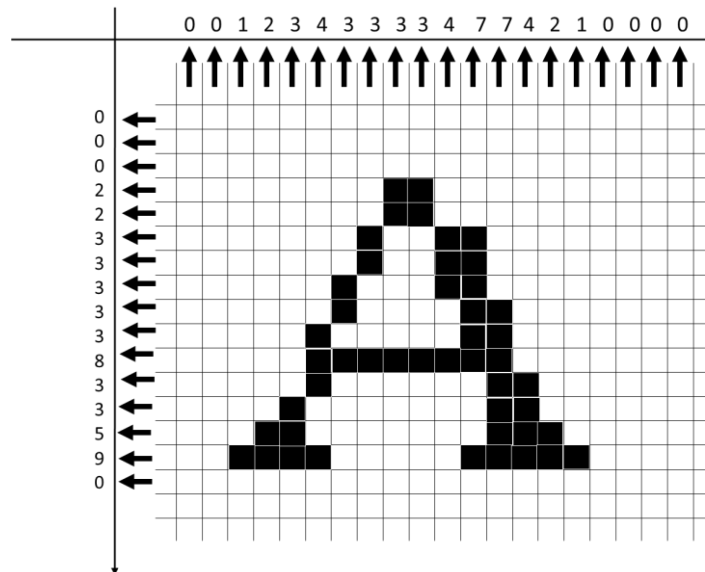


Figure I. 2: Groupe de pixels formant la lettre A

I.3.3. La résolution

La résolution d'une image correspond au niveau de détail qui va être représenté sur cette image. C'est le nombre de pixels par unité de longueur dans l'image à numériser. Elle est en dpi (dots per inch) ou en ppp (points par pouce).

Plus le nombre de pixels est élevé par unité de longueur de l'image à numériser, plus la quantité qui décrit l'image est importante et plus la résolution est élevée [13].

I.3.4. Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

I.3.5. Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur [14].



Figure I. 3: (a) Image sans bruit. (b) Image avec bruit

I.3.6. La luminance

C'est 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, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes);
- Un bon contraste: il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites [15].

I.4. Représentation d'une l'image

Il existe deux méthodes fondamentalement différentes de représentation et de codage des images: la méthode *matricielle* (*Bitmap*) et la méthode *vectorielle*. Un fichier graphique peut appartenir à l'une ou l'autre de ces méthodes, ou être une combinaison des deux.

Les images de type matricielle (bitmap)

Les images affichées sur un écran sont des images matricielles, appelées bitmap. Une image bitmap est composée en mode point appelé **pixel** et codé dans la mémoire vidéo de l'ordinateur par leurs coordonnées spatiales et leur couleur.

Les images de type vectorielle

La description vectorielle d'une image consiste en une description géométrique: le document numérisé prend donc la forme d'une suite de formules mathématiques décrivant les formes élémentaires constituant l'image (carrés, rectangles, ellipses, cercles, courbes, etc.).

I.5. Types d'images

On distingue trois types d'images :

- Binaire : 2 couleurs (arrière-plan et avant-plan).
- Monochrome : variations d'une même teinte.
- Polychrome (couleur): " vraies " couleurs [11].

I.5.1. Images binaires (en noir et blanc)

Les images binaires sont les plus simples. Bichromes (dont les pixels ne peuvent avoir que la valeur 0 et 1). Le 0 correspond à un pixel noir et 1 à un pixel blanc. Le niveau de gris est donc codé sur un seul bit.



Figure I. 4: Image en noire et blanc.

I.5.2. Images en niveaux de gris (Monochromes)

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 au blanc en passant par un nombre fini de niveaux intermédiaires. Donc 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. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par 1 bit, mais par 1 octet. 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 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 [11].



Figure I. 5: Image monochrome.

I.5.3. Images en couleur (Polychromes)

Elle est obtenue par la combinaison de trois couleurs dites primaires : rouge, vert et bleu (RVB). Chaque couleur est codée comme une image à niveaux de gris, avec des valeurs allant de 0 à 255. Pour $R=V=B=0$, nous aurons un noir pur, et pour $R=V=B=255$, nous aurons un

blanc pur. La représentation des images couleur se fait donc soit par une image dont la valeur du pixel est une combinaison linéaire des valeurs des trois composantes couleurs, soit par trois images distinctes représentant chacune une composante couleur, on distingue généralement deux types d'images [14]:

- Images 24 bits
- Images à palettes



Figure I. 6: Image polychrome (couleur).

I.6. Les différents formats d'image [16]

Le tableau suivant représenté Les différents formats d'image :

Nom du format	Type d'image	Compression des données	Nombre de couleurs supportées	Affichage progressif	Animation
TIFF <i>Tagged Image File Format</i>	Matriciel	Compression ou pas avec ou sans perte	De monochrome à 16 millions.	Non	Non
JPEG <i>Joint Photographic Expert Group</i>	Matriciel	Oui, avec perte.	JPEG (16 millions)	Oui	Non
JPEG 2000 <i>Joint Photographic Expert Group 2000</i>	Matriciel	Oui, avec perte.	JPEG 2000 (32 millions)	Oui	Oui
GIF <i>Graphics Interchange Format</i>	Matriciel	Oui, sans perte.	256 maxi (palette)	Oui	Oui
PNG <i>Portable Network Graphic</i>	Matriciel	Oui, sans perte.	Palettisé (256 couleurs ou moins) ou 16 millions	Oui	Non

Tableau I. 1: Les différents formats d'images.

I.7. Définition de la compression

La compression de données, d'une manière générale c'est l'ensemble des méthodes et règles qui permettent de réduire le volume d'une donnée sans perdre les informations essentielles. D'une manière plus simple, c'est une technique où on emploie une paire de fonctions, une d'elle a pour but de compresser les données et l'autre de les décompresser.

L'objectif de la compression de données est de représenter l'information sous une forme plus compacte. Les données compressées occupent moins d'espace que les données originales: le nombre de bits utilisés pour représenter les données est réduit. Les données peuvent être compressées avec ou sans perte. Nous nous sommes intéressés exclusivement à la compression de données sans perte [17].

I.7.1. Différence entre la compression avec perte et la compression sans perte

La compression de données est une méthode permettant de réduire la taille des données sans perte significative d'informations. La compression avec perte et la compression sans perte sont les deux termes largement catégorisés sous les méthodes de compression de données. La principale différence entre la compression avec perte et la compression sans perte est que la compression avec perte produit une correspondance proche des données après la décompression alors que la méthode sans perte crée des données d'origine exactes.

D'autre part, la compression avec perte réduit un fichier en éliminant définitivement certaines informations, en particulier des informations redondantes. La compression avec perte est généralement utilisée pour la vidéo et le son, où une certaine quantité de perte d'informations ne sera pas détectée par des utilisateurs.

I.8. Définition des critères utilisés en compression

I.8.1. Taux de compression

Le processus de compression d'image vise à représenter l'image avec une séquence de bits appelée Bistream et à réduire sa taille autant que possible. Le taux de compression est un outil de mesure pour évaluer le degré de compression obtenu. Il est le rapport entre le nombre de bits utilisés par l'image originale et le nombre de bits utilisés par l'image compressée. Il se calcule par conséquent comme suit:

$$\tau = \frac{\text{nombre de bit utilisé pour représenter l'image originale}}{\text{nombre de bit utilisés par l'image compressée}} \quad (\text{I.1})$$

$$\tau = \frac{N \times M \times B}{\text{nombre de bit du bistream}} \quad (\text{I.2})$$

Où :

N, M : représentent respectivement le nombre de lignes et le nombre de colonnes de l'image.

B: désigne le nombre de bits utilisés pour la représentation d'un pixel.

I.8.2. Le débit

Le débit, qui est une autre évaluation fréquemment employée, est déterminé à partir du nombre moyen de bits par pixel (bpp) de l'image compressée, il est calculé de la manière suivante:

$$\text{débit} = \frac{\text{nombre de bits par pixel dans l'image originale}}{\tau} \text{ bits par pixel (bpp)} \quad (\text{I.3})$$

$$\text{débit} = \frac{\text{nombre de bits du Bistream}}{N \times M} \quad (\text{I.4})$$

I.8.3. Critères de qualités

Pour la compression sans perte, la seule chose qui nous intéresse est d'obtenir le meilleur taux de compression (ou débit) possible. La compression avec perte vise à réduire le nombre de bits nécessaires pour représenter une image avec un niveau de distorsion acceptable. En plus de chercher à obtenir un taux de compression plus élevé, nous nous intéresserons également à l'évaluation de la différence entre l'image originale et l'image reconstruite (décodée).

I.9. Quantité d'information et l'entropie

La théorie d'information permet d'évaluer la quantité d'information dans une image. Pour cela, chaque point d'une image est considéré comme une variable aléatoire.

Soit P un point d'une image en niveau de gris. C'est une variable aléatoire dont les valeurs sont des entiers de l'intervalle [0 255].

Soit $p(n_i)$, la probabilité pour que le niveau de gris en P soit n_i .

→ La quantité d'information est donnée par :

$$I(n_i) = \log_a \left(\frac{1}{p(n_i)} \right) = -\log_2(p(n_i)) \quad (\text{I.5})$$

$I(n_i)$: est l'incertitude (quantité d'information).

→ On définit l'entropie comme suit:

$$H(p) = \sum_{i=1}^N p(n_i) I(n_i) = - \sum_{i=1}^N p(n_i) \log_2(p(n_i)) \quad (\text{I.6})$$

L'entropie $H(p)$ est exprimée en bits/symbole (bits/pixel pour les images).

I.10. La compression JPEG

La compression JPEG (Joint Photographie Experts Group) est une technique efficace pour la compression d'images.

I.10.1. Principe de compression du JPEG

La norme JPEG décrit le format des données compressées et le schéma codage et de décodage. Les figures suivantes illustrent les étapes de compression et décompression.

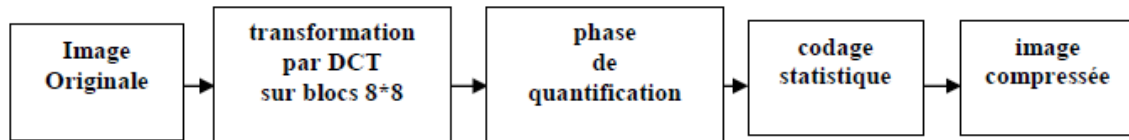


Figure I. 7: Les étapes de la Compression

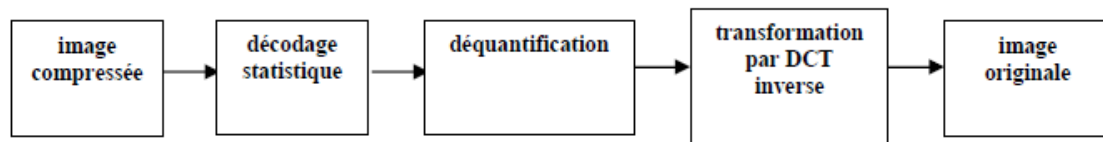


Figure I. 8: Les étapes de la décompression

I.10.2. Utilité et caractéristiques de la compression JPEG

- JPEG doit pouvoir compresser n'importe quel type d'images réelles. (Peut gérer différentes formes d'expression des couleurs comme RGB, CMJ ou les niveaux de gris).
- L'algorithme doit être implémenté sans trop de problèmes sur une grande gamme de CPUs, et sur des cartes spécialisées.
- Le codage doit pouvoir être séquentiel, progressif et sans pertes.
- JPEG est destiné à des images de type photographique. Les images composées de vecteurs, de droite ou d'objet géométrique, ne seront pas compressées avec une excellente efficacité [18].

I.11. Conclusion

Ce chapitre présente dans un premier temps les généralités et les connaissances de base sur les images. Dans un deuxième temps, nous avons introduit les concepts et les outils liés au domaine de la compression d'images et de données. Enfin, les outils utiles pour apprécier et comparer la performance des algorithmes de compression ont été présentés.

Chapitre II

Techniques de codage sans perte

II.1. Introduction

II.2. Définition du codage

II.3. Les différentes méthodes de compression sans perte

II.4. Codage à base de dictionnaire

II.5. Le codage prédictif

II.6. Histogramme (Cameraman)

II.7. Application à la compression d'images

II.8. Algorithmes basées sur une étape de codage prédictif

II.9. Conclusion

II.1. Introduction

En exploitant la redondance informationnelle du signal (ou de l'image), les méthodes de compression visent à réduire la longueur moyenne, exprimée bits par échantillon, du signal à stocker ou à transmettre. Ce chapitre expose les bases des principaux processus de codage sans perte, à savoir Shannon-Fano, Huffman, Arithmétique, RLE, LZW, etc. Nous présentons également quatre variantes basées sur les deux techniques Huffman et Arithmétique pour application en compression d'images sans perte. Les deux premières variantes appliquent les codages Huffman et arithmétique au domaine spatial directement sur l'image tandis que les deux autres variantes appliquent les codages Huffman et Arithmétique une fois le codage prédictif est effectué.

II.2. Définition du codage

Le codage est une étape essentielle de la compression, il est une méthode de transformation qui convertit la représentation d'une information en une autre. [17]. Il existe plusieurs techniques de codage telles que Shannon-Fano, Huffman, etc.

II.3. Les différentes méthodes de compression sans perte

II.3.1. Le codage de Shannon-Fano

Le codage de Shannon-Fano a été proposé en 1948 par Claude Shannon dans l'article 'A Mathematical Theory of Communication' [19]. Shannon du laboratoire Bells et R. M. Fano du MIT ont développé à peu près en même temps une méthode de codage basée sur de simples connaissances de la probabilité d'occurrence de chaque symbole dans le message [13].

Le procédé de Shannon-Fano construit un arbre descendant à partir de la racine, par divisions successives. Le classement des fréquences se fait par ordre décroissant, ce qui suppose une première lecture du fichier et la sauvegarde de l'en-tête.

Le principe est le suivant :

- Classer les n fréquences non nulles $\{f_i\}$ par ordre décroissant.
- Répartir la table des fréquences en deux sous tables de fréquences proches.

Poursuivre l'arborescence jusqu'à ce que toutes les fréquences soient isolées.

- Attribuer dans l'arborescence le bit 0 à chaque première sous table.
- Attribuer aux symboles les codes binaires correspondant aux bits de description 1 de l'arborescence.

Exemple : Soit la figure suivante qui représente la fréquence d'apparition des symboles d'un Alphabet A= A, B, C, D, E [7]. Les statistiques sont extraites à partir d'un texte :

Symbole	A	B	C	D	E
Apparitions	15	7	6	6	5
Fréquence	15/39	7/39	6/39	6/39	5/39

Tableau II. 1 : Fréquences d'apparition des symboles de l'alphabet

L'application de cet algorithme nous fournit l'arbre suivant [17] :

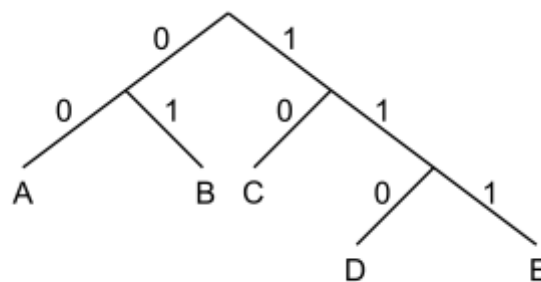


Figure II. 1: Algorithme de Shannon-Fano sur l'alphabet

A partir de l'arbre on peut extraire le code de chaque symbole. Ce qui nous donne le tableau suivant [17] :

Symbole	Mot de code
A	00
B	01
C	10
D	110
E	111

Tableau II. 2: Code de Shannon-Fano pour l'alphabet

Avantages et inconvénients

- **Avantages**

Produit des codes préfix.

Réduction importante dans la taille du fichier à compresser.

- **Inconvénients**

La taille des codes augmente avec le nombre de symboles qui existent dans l'alphabet utilisé.

II.3.2. Le codage de Huffman

En 1952, David Huffman inventa une nouvelle méthode de compression appelée compression à Arbre de Huffman [20]. Le codage de Huffman crée des codes à longueurs variables sur un nombre entier de bits [21].

Le codage de Huffman est basé sur la fréquence d'occurrence d'un élément de données (pixel dans les images). Le principe est d'utiliser un nombre inférieur de bits pour coder les données qui se produit plus fréquemment. Les codes sont stockés dans un dictionnaire de code qui peut être construit pour chaque image ou un ensemble d'images.

Dans tous les cas, le dictionnaire de code ainsi que des données codées doivent transmettre pour permettre le décodage [22].

Pour une source x d'entropie $H(x)$, la longueur moyenne L_H d'un mot de code obtenu par codage de Huffman vérifie la relation suivante:

$$H(x) \leq L_H < H(x) + 1 \quad (\text{II.1})$$

En général, les codes de Huffman sont utilisés au second étage de compression, conjointement avec d'autres techniques de codage. Ces codes permettent des applications de compression telles que la compression d'images sans perte, la compression du texte, la compression du son [1].

II.3.2.1. Avantages

Le codage de Huffman offre une compression caractère par caractère optimale, c'est-à-dire qu'on ne peut pas obtenir des codes binaires plus courts pour les caractères. Il a aussi l'avantage d'être facile à implémenter par programmation, et le temps d'exécution est plutôt rapide.

Exemple :

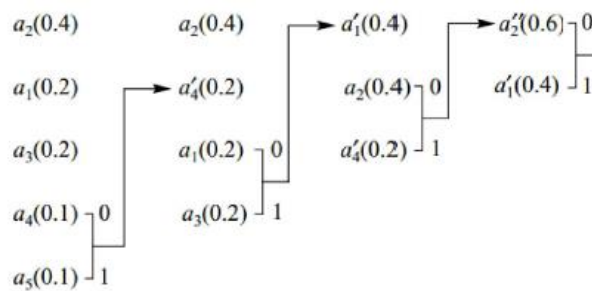


Figure II. 2: La procédure de codage de Huffman à variance minimale

Symbole	Probabilité	Mot de code
a_1	0.2	10
a_2	0.4	00
a_3	0.2	11
a_4	0.1	010
a_5	0.1	011

Tableau II. 3 : Code de Huffman à variance minimale.

II.3.3. Le codage Arithmétique

Le codage Arithmétique [23] est un codage utilisant un modèle statistique, c'est-à-dire que plus un caractère est représenté, moins il faudra de bits pour le coder.

Il s'agit d'un cousin du codage de Huffman qui cependant reste toujours plus efficace que ce dernier (sauf dans le cas particulier où tous les poids des feuilles/nœuds/racines de l'arbre de Huffman sont des puissances de 2). Il est aussi plus simple à implémenter.

L'avantage que possède le codage Arithmétique sur le codage de Huffman est que ce dernier va coder un caractère sur un nombre entier de bits (il ne peut coder sur 1.5 bits) là où le codage arithmétique le peut. Par exemple, si un caractère est représenté à 90%, la taille optimale du code du caractère serait de 0.15 bit, alors que Huffman coderait sûrement ce symbole sur 1 bit, soit 6 fois trop.

Le codage de Huffman permet de coder toujours un symbole (d'une source) avec un nombre entier de bits. Le codage Arithmétique est alors une méthode plus performante, largement répondu. L'un de ses avantages est qu'il n'attribue pas un mot de code à chaque symbole mais

à un ensemble de symboles (message en entier) en lui associant un nombre en virgule flottante appartenant à l'intervalle $[0,1[[1]$.

II.3.3.1. Procédure du codage Arithmétique

- Calculer la probabilité associée à chaque symbole dans la chaîne à coder.
- Associer à chaque symbole un sous intervalle proportionnel à sa probabilité, dans l'intervalle $[0,1]$ (l'ordre de rangement des intervalles sera mémorisé car il est nécessaire au décodeur)
- Initialiser la limite inférieure de l'intervalle de travail à la valeur 0 et la limite supérieure à la valeur 1.

Tant qu'il reste un symbole dans la chaîne à coder:

- largeur = limite supérieure - limite inférieure.
 - limite inférieure = limite inférieure + largeur x (limite basse du sous intervalle du symbole).
 - limite supérieure = limite inférieure + largeur x (limite haute du sous intervalle du symbole).
- La limite inférieure code la chaîne de manière unique [16].

II.4. Codage à base de dictionnaire

Les méthodes telles que celle de Huffman, Shannon-Fano sont basées sur des modèles statistiques, et la probabilité d'occurrence joue particulièrement un rôle important.

Dans les techniques de compression de données basées sur un dictionnaire, un symbole ou une chaîne de symboles générés à partir d'un alphabet source est représentée par un index d'un dictionnaire construit à partir de l'alphabet source. Un dictionnaire est une liste de symboles et de chaînes de symboles. Le codage à base du dictionnaire est largement utilisé dans le codage du texte. La stratégie du codage du dictionnaire consiste à construire un dictionnaire contenant des symboles et les chaînes de symboles qui se produisent fréquemment. Lorsqu'un symbole (ou une chaîne de symbole) est rencontré et qu'il existe dans le dictionnaire, il est encodé avec un index du dictionnaire. Sinon, si le symbole (la chaîne) ne figure pas dans le dictionnaire, le symbole (ou la chaîne de symboles) est codé de manière moins efficace.

II.4.1. RLE (Run-length-encoding)

L'encodage RLE consiste à remplacer toutes les suites de caractères ou bits pareils par un nombre représentant le nombre de répétitions du caractère ou bits suivit de Celui-ci. Généralement, pour moins de gaspillage d'espace, on ne remplace une chance que si le nombre de répétition du caractère ou du bit est supérieur à deux.

a. But

Cet algorithme élide les répétitions successives de caractères.

b. Algorithme de compression

Recherche des caractères répétés plus de n fois (n fixé par l'utilisateur).

Remplacement de l'itération de caractères par:

1. un caractère spécial identifiant une compression.
2. le nombre de fois où le caractère est répété.
3. le caractère répété [16].

c. Algorithme de décompression

Durant la lecture du fichier compressé, lorsque le caractère spécial est reconnu, on effectue l'opération inverse de la compression tout en supprimant ce caractère spécial.

Exemple : AAAAARRRRRROLLLLBBBBBUUTTTTTT.

On choisit comme caractère spécial : @ et comme seuil de répétition : 3.

Après compression : @5A@6R@4L@5BUU@6T gain : 11 caractères soit 38% [24].

d. Utilité

Essentiellement pour la compression des images (car une image est composée de répétitions de pixels, de couleur identique, codés chacun par un caractère).

e. Caractéristiques de compression

- algorithme très simple.
- taux de compression relativement faible (40%).

C'est une méthode utilisée par de nombreux formats d'images (BMP, PCX, TIF) [24].

II.4.2. LZW (Lempel Ziv Welch)

L'algorithme de classe LZ77 utilise un horizon pour trouver les concordances qui aideront à la compression. Mais cette méthode a été améliorée avec l'arrivée de l'algorithme LZ78 qui utilise une structure de données auxiliaire pour stocker la séquence répétée, cette structure est appelée un dictionnaire. Cette fois pour représenter une séquence répétée on utilise l'index de cette séquence dans le dictionnaire mais pas la position. On aura alors des tuples sous forme <index, &> ou & représente le caractère qui suit la séquence répétée [25].

LZW est capable de travailler avec n'importe quel type de données. Il est rapide en compression et décompression et ne nécessite pas d'opération à virgule flottante. De par le fait qu'il encode au niveau bit et non au niveau de l'octet, il ne se soucie pas du processeur, et de la manière dont il code les informations.

LZW utilise la substitution des motifs en se basant sur un dictionnaire construit au fil de la compression.

Le flot d'information à compresser est découpé en chaînes d'octets. Chaque chaîne est comparée au dictionnaire. Si elle n'est pas présente, elle est stockée. Elle est ensuite écrite dans le flot de sortie compressé.

- Quand une chaîne, déjà rencontrée, apparaît dans le flot, elle est codée et transmise si elle a une longueur inférieure au plus grand mot du dictionnaire.
- Pour le décodage, le logiciel reconstruit en fait le dictionnaire dans le sens inverse. Il n'est pas nécessaire ainsi de transmettre le dictionnaire. Bien souvent, pour la compression et la décompression, le dictionnaire est initialisé avec les 256 valeurs de la table ASCII. Ainsi, tous les codeurs et décodeurs LZW initialisent leurs dictionnaires de la même méthode.
- Le format TIFF effectue une concaténation de points. Les données sont ensuite soumises à la compression LZW. Ceci peut parfois produire quelques problèmes si l'image ne possède pas un nombre pair de point.
- Le format GIF quant à lui code chaque point sur un octet, puis soumet la suite d'octets à la compression LZW.

a. But

Cet algorithme réduit la taille des chaînes de caractères (c'est-à-dire les mots) récurrents.

b. Algorithme

LZW a introduit l'idée que seul le dictionnaire initial (le dictionnaire initial contient tous les symboles individuels (l'alphabet) de la source) doit être transmis pour permettre le décodage: Le décodeur est en mesure de construire le reste du tableau à partir de la séquence codée.

→ L'algorithme de compression LZW peut être résumé comme suit:

```
w = NIL;
while( lire un caractère k )
{
ifwk existe dans le dictionnaire
w = wk;
else
```

ajouter wk au dictionnaire;

Transmettre le code pour w;

w = k;

}

Transmettre le code pour w

Exemple: la chaîne à coder est "HAP-HA-KAB-HAA-KS".

W	K	Output	Index	Symbol
NIL	H			
H	A	4	8	HA
A	P	2	9	AP
P	-	6	10	P-
-	H	1	11	-H
H	A			
HA	-	8	12	HA-
-	K	1	13	-K
K	A	5	14	KA
A	B	2	15	AB
B	-	3	16	B-
-	H			
-H	A	11	17	-HA
A	A	2	18	AA
A	-	2	19	A-
-	K			
-K	S	13	20	-KS
S		7		

Tableau II. 4: Codage LZW.

→ L'algorithme de décompression est comme suit:

lire un caractère k;

transmettre k;

w = k;

```

while (lire un caractère k )
/* k peut être un caractère ou un code. */
{
entry = entrée du dictionnaire pour k;
transmettre entry;
ajouter w + entry[1] au dictionnaire;
w = entry;
}

```

Exemple: Le décodage de l'exemple précédent est comme suit:

La chaîne à décoder est : "HAP-HA-KAB-HAA-KS".

La chaîne décodée est identique à la chaîne d'entrée.

W	K	Output	Index	Symbol
NIL	4	H		
H	2	A	8	HA
A	6	P	9	AP
P	1	-	10	P-
-	8	HA	11	-H
HA	1	-	12	HA-
-	5	K	13	-K
K	2	A	14	KA
A	3	B	15	AB
B	11	-H	16	B-
-H	2	A	17	-HA
A	2	A	18	AA
A	13	-K	19	A-
-K	7	S	20	-KS

Tableau II. 5: Décodage LZW.

II.5. Le codage prédictif

Le codage prédictif est une technique, qui permet obtenir une bonne compression sans nécessiter une surcharge de calcul significative, peut être soit sans perte, soit avec perte.

Cette approche est basée sur l'élimination des redondances des pixels voisins dans le domaine spatial, par extraction et codage uniquement de la nouvelle information qui apparaît dans chaque pixel. La nouvelle information d'un pixel est définie comme la différence entre la valeur réelle et prédite de ce même pixel [10].

II.5.1. Codage prédictif sans perte

La figure II.3 montre les composants de base d'un système de codage prédictif sans perte. Le système se compose d'un codeur et d'un décodeur, chacun contenant un prédicteur identique. Au fur et à mesure que des échantillons successifs de signal d'entrée à temps discret, $f(n)$, sont introduits dans le codeur, le prédicteur génère la valeur prédite de chaque échantillon sur la base d'un nombre spécifié d'échantillons passés. La sortie du prédicteur est ensuite arrondie à l'entier le plus proche, $\hat{f}(n)$, qui est utilisé pour former la différence ou l'erreur de prédiction.

$$e(n) = f(n) - \hat{f}(n) \quad (\text{II.2})$$

Cette différence (ou erreur) est codée à l'aide d'un code de longueur variable (par le codeur de symboles) pour générer l'élément suivant du flux de données compressé (Bitstream). Le décodeur de la Fig. (II.3) reconstruit $e(n)$ à partir des mots de code de longueur variable reçus et effectue l'opération inverse suivante pour décompresser ou reconstruire la séquence d'entrée originale.

$$f(n) = e(n) + \hat{f}(n) \quad (\text{II.3})$$

Diverses méthodes locales, globales et adaptatives peuvent être utilisées pour générer $\hat{f}(n)$. Dans de nombreux cas, la prédiction est formée comme une combinaison linéaire de m échantillons précédents. C'est-à-dire :

$$\hat{f}(n) = \text{round} \left[\sum_{i=1}^m \alpha_i f_{n-i} \right] \quad (\text{II.4})$$

Où: m est l'ordre du prédicteur linéaire, round est une fonction utilisée pour désigner l'opération d'arrondi au plus proche entier, et les α_i pour $i = 1, 2, \dots, m$ sont des coefficients de prédiction. Si la séquence d'entrée de la figure II. 3 (a) est considérée comme des échantillons d'une image,

les $f(n)$ dans les équations (II.2) à (II.4) sont des pixels, et les m échantillons utilisés pour prédire la valeur de chaque pixel proviennent de la ligne de balayage courante (appelé codage prédictif linéaire 1-D). Ainsi, pour le codage d'images prédictif linéaire 1-D, l'équation (II.4) peut s'écrire comme suit [10]:

$$\hat{f}(x, y) = \text{round} \left[\sum_{i=1}^m \alpha_i f(x, y - i) \right] \quad (\text{II.5})$$

Où chaque échantillon est maintenant exprimé explicitement comme une fonction des coordonnées spatiales de l'image d'entrée, x et y . Notez que la prédiction $\hat{f}(x, y)$ (la prédiction linéaire 1-D) est une fonction des pixels précédents sur la ligne actuelle uniquement [10]. Dans le codage prédictif 2-D, la prédiction est fonction des pixels précédents dans un balayage de gauche à droite et de haut en bas d'une image.

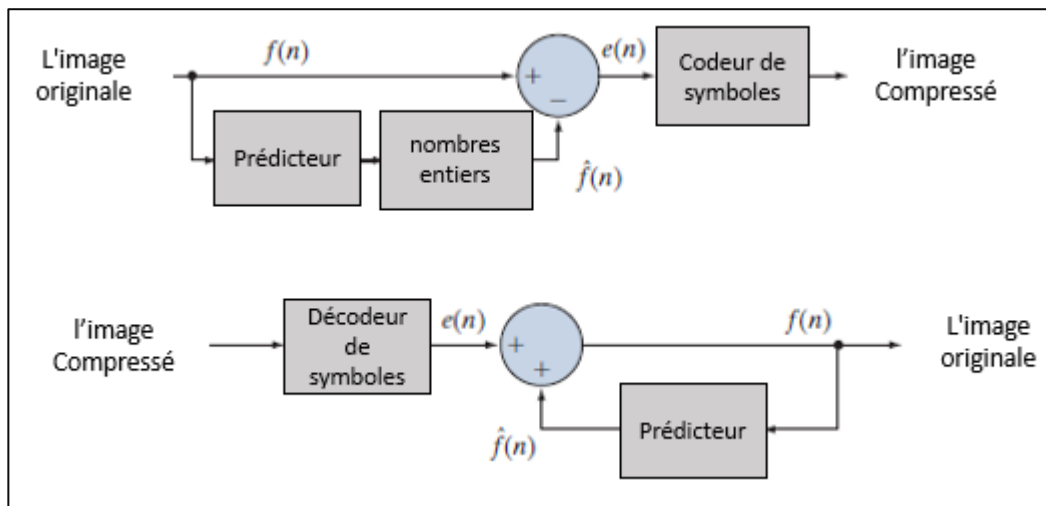


Figure II. 3: Un modèle de codage prédictif sans perte : codeur et décodeur.

II.6. Histogramme (Cameraman)

Les figures II.4 et II.5 suivantes montrent l'histogramme d'une image originale et l'image de l'erreur de prédiction :

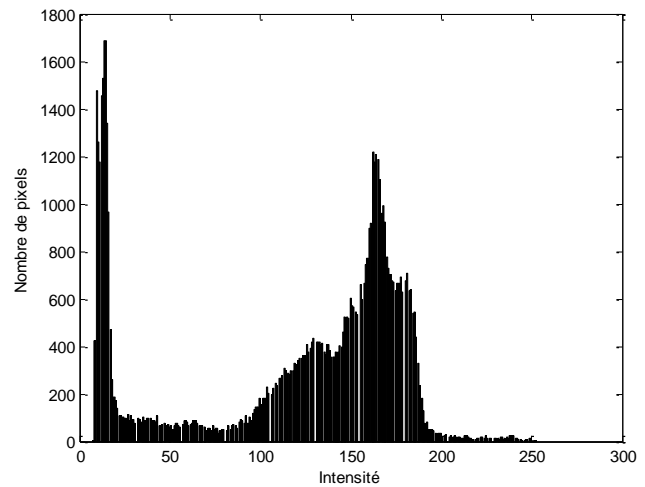


Figure II. 4: Un histogramme d'une image originale.

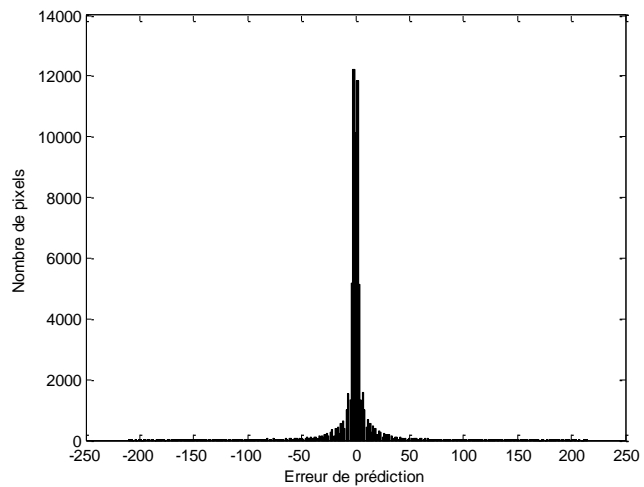
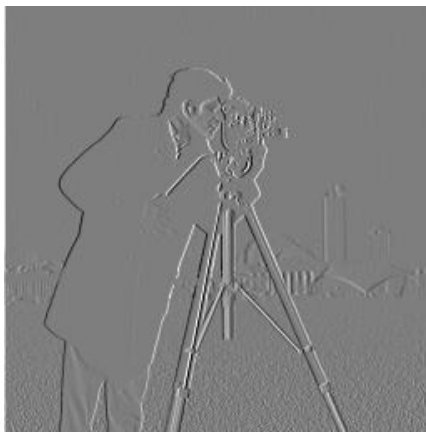


Figure II. 5: Histogramme de l'erreur de prédiction.

II.7. Application à la compression d'images

Le schéma de compression/décompression sans perte des images dans le domaine spatial est illustré dans les figures ci-dessous [16]: (l'image originale), (l'image compressée).



Figure II. 6: Schéma de compression.



Figure II. 7: Schéma de décompression.

II.7.1. Algorithme du codage de Huffman (CH)

L'algorithme de compression/décompression utilisant le codage de Huffman est comme suit:

- Lire l'image à compresser.
- Transformer l'image en un vecteur.
- Calculer l'Alphabet, les fréquences et les probabilités.
- Effectuer le codage de Huffman.
- Effectuer le décodage de Huffman.
- Transformer le vecteur en une matrice.
- Calculer l'entropie, le taux de compression et le débit en Bpp (Bits par pixel).
- Afficher la taille de l'image originale en Bits et en Kilo octets.
- Afficher la taille de l'image compressée en Bits et en Kilo octets.
- Afficher l'entropie, le débit et le taux de compression.

II.7.2. Algorithme du codage Arithmétique (CA)

L'algorithme de compression/décompression utilisant le codage Arithmétique est le suivant:

- Lire l'image à compresser.
- Transformer l'image en un vecteur.
- Calculer l'Alphabet, les fréquences et les probabilités.
- Effectuer le codage d'Arithmétique.
- Effectuer le décodage d'Arithmétique.
- Transformer le vecteur en une matrice.

- Calculer l'entropie, le taux de compression et le débit en Bpp (Bits par pixel).
- Afficher la taille de l'image originale en Bits et en Kilo octets.
- Afficher la taille de l'image compressée en Bits et en Kilo octets.
- Afficher l'entropie, le débit et le taux de compression.

II.8. Algorithmes basés sur une étape de codage prédictif

Le schéma synoptique de compression/décompression sans perte des images à base du codage prédictif est illustré dans les figures ci-dessous:

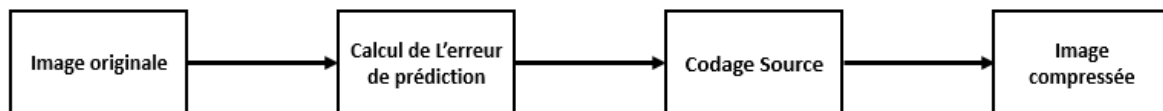


Figure II. 8: Schéma de compression.

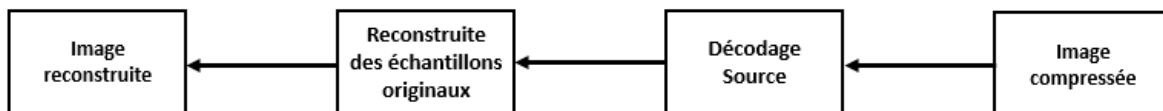


Figure II. 9: Schéma de décompression.

II.8.1. Algorithme du Codage de Huffman associé au codage Prédictif (CHP)

L'algorithme de compression/décompression utilisant le codage de Huffman combiné avec le codage prédictif est défini de la manière suivante:

- Lire l'image à compresser
- Calcul de l'erreur de prédiction.
- Transformer l'image en un vecteur.
- Calculer l'Alphabet, les fréquences et les probabilités.
- Effectuer le codage de Huffman.
- Effectuer le décodage de Huffman.
- Transformer le vecteur en une matrice.
- Reconstituée des échantillons originaux.
- Calculer l'entropie, le taux de compression et le débit en Bpp (Bits par pixel).
- Afficher la taille de l'image originale en Bits et en Kilo octets.
- Afficher la taille de l'image compressée en Bits et en Kilo octets.
- Afficher l'entropie, le débit et le taux de compression.

II.8.2. Algorithme du codage Arithmétique associé au codage Prédicatif (CAP)

L'algorithme de compression/décompression utilisant le codage d'Arithmétique combiné avec codage prédictif est présenté ci-dessous:

- Lire l'image à compresser.
- Calcul de l'erreur de prédiction.
- Transformer l'image en un vecteur.
- Calculer l'Alphabet, les fréquences et les probabilités.
- Effectuer le codage Arithmétique.
- Effectuer le décodage Arithmétique.
- Transformer le vecteur en une matrice.
- Reconstruite des échantillons originaux.
- Calculer l'entropie, le taux de compression et le débit en Bpp (Bits par pixel).
- Afficher la taille de l'image originale en Bits et en Kilo octets.
- Afficher la taille de l'image compressée en Bits et en Kilo octets.
- Afficher l'entropie, le débit et le taux de compression [16].

II.9. Conclusion

Dans ce chapitre, différentes techniques du codage source sans perte ont été explorées. Une présentation de quatre variantes deux techniques de codage Huffman et Arithmétique est également effectuée pour application en compression d'images sans perte. Les deux premières variantes appliquent les codages Huffman et Arithmétique au domaine spatial directement sur l'image tandis que les deux autres variantes appliquent les codages Huffman et Arithmétique en conjonction avec le codage prédictif.

Chapitre III

Simulations et résultats

III.1. Introduction

III.2. Les données de test et leurs caractéristiques

III.3. Machine PC

III.4. Evaluation de la performance en termes de débit (en Bpp)

III.5. Evaluation de la performance en termes de complexité de calcul (temps de codage et décodage)

III.6. Comparaison de la performance des techniques de codage utilisant le codage prédictif, les techniques CHP et CAP en termes de débit (en Bpp).

III.7. Conclusion

III.1. Introduction

Ce chapitre traite de l'évaluation des performances de compression sans perte de deux techniques de codage de source, à savoir le codage de Huffman et le codage arithmétique, appliquées séparément et conjointement avec le codage prédictif. Pour ce faire, les performances de compression sans perte des variantes nommées: CH (Codage de Huffman), CHP (codage de Huffman associé au codage prédictif), CA (Codage Arithmétique) et CAP (codage Arithmétique associé au codage prédictif) sont évaluées et comparées en termes de débit et de temps de calcul, à l'aide d'un ensemble d'images en niveaux de gris de dimension variable avec des caractéristiques bien distinctes.

III.2. Les données de test et leurs caractéristiques

- Cameraman, dimension 256 x 256, taille 64 Ko.
- Sailboat, dimension 512 x 512, taille 256 Ko.
- GrandPhare, dimension 768 x 512, taille 384 Ko.
- Airplane, dimension 1024 x 1024, taille 1.00 Mo.
- Flower, dimension 512 x 480, taille 240 Ko.
- Peppers, dimension 512 x 512, taille 256 Ko.
- Baboon, dimension 512 x 512, taille 256 Ko.
- Bikes, dimension 768 x 512, taille 384 Ko.
- Male, dimension 1024 x 1024, taille 1.00 Mo.
- Couple, dimension 512 x 512, taille 256 Ko.

III.3. Les caractéristiques de Machine PC

- Edition Windows: Windows 10 Professionnel
- Type de système: Système d'exploitation 64 bits.
- Processeur: Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz 2.00 GHz
- Mémoire installée (RAM): 4.00 Go.
- Environnement de travail dans Matlab R2013b.

III.4. Evaluation de la performance en termes de débit (en Bpp)

III.4.1. Evaluation de la performance des techniques de codage: CH (codage de Huffman) et CHP (codage de Huffman associé au codage prédictif)

Image	Taille de l'image originale	Codage de Huffman (CH)			Codage de Huffman associé au codage Prédicatif (CHP)		
		Entropie	Débit	Taille de l'image compressée	Entropie	Débit	Taille de l'image compressée
Cameraman	524288 Bits 64 KO	7.0097	7.0448	461689 Bits 56.3585 KO	5.0551	5.0787	332840 Bits 40.6299 KO
Sailboat	2097152 Bits 256 KO	7.2659	7.2951	1912365 Bits 233.4430KO	5.3879	5.4116	1418623 Bits 173.1718KO
GrandPhare	3145728 Bits 384 KO	7.0398	7.0702	2780125 Bits 339.3707KO	5.0441	5.0758	1995905 Bits 243.6407 KO
Airplane	8388606 Bits 1024 KO	5.6415	5.6767	5952458 Bits 726.6184 KO	4.7336	4.7564	4987454 Bits 608.8201 KO
Flower	1966080 Bits 240 KO	7.1913	7.2241	1775403 Bits 216.7240 KO	4.1402	4.1626	1023010 Bits 124.8792 KO
Peppers	2097152 Bits 256 KO	7.3744	7.3975	1939215 Bits 236.7206 KO	4.8780	4.9042	1285595 Bits 156.9330 KO
Baboon	2097152 Bits 256 KO	7.1391	7.1673	1878856 Bits 229.3525 KO	6.1471	6.1698	1617369 Bits 197.4327 KO
Bikes	3145728 Bits 384 KO	7.3720	7.4055	2911949 Bits 355.4625 KO	5.9007	5.9426	2336743 Bits 179.9675 KO
Male	8388608 Bits 1024 KO	7.5237	7.5486	7915331 Bits 966.2269 KO	5.2929	5.3193	5577700 Bits 680.8716 KO
Couple	2097152 Bits 256 KO	7.2010	7.2327	1896014 Bits 231.4470 KO	5.0607	5.0986	1336580 Bits 163.1567 KO
Moyenne	/	7.0758	7.1063	/	5.1640	5.1920	/
Taux de compression moyen	/	1.1306	1.1257	/	1.5491	1.5408	/

Tableau III. 1: Tableau comparatif des performances de compression sans perte des techniques de codage: CH et CHP.

Dans le cadre de cette étude, nous avons employé un ensemble d'images en niveaux de gris de différentes tailles : 256 x 256, 512 x 512, 768 x 512 et 1024 x 1024, pour mettre une évaluation valide de l'efficacité des deux techniques de codage : CH (codage de Huffman) et CHP (codage de Huffman associé au codage prédictif). L'entropie et le débit exprimés en bits par pixel (bpp) sont des outils de référence souvent utilisés comme mesure comparative de la performance de compression sans perte. Le tableau (III.1) et figure (III.1) présentent les résultats obtenus. Pour permettre une comparaison éclairante, les tailles des images compressées par les deux techniques sont mesurées en bits et en kilo-octets (K.O) comme l'illustre dans ce tableau.

Il ressort des données fournies dans le tableau (III.1) et figure (III.1) que la technique du codage prédictif combinée avec le codage de Huffman réalise la meilleure performance. On constate d'abord la diminution significative de l'entropie avec l'utilisation du codage prédictif, entraînant ainsi une diminution importante du débit et de la taille des images compressées.

Par rapport au codage de Huffman, le codage prédictif combiné avec le codage de Huffman réalise des gains significatifs qui varient entre 0.92 Bpp et 3.06 Bpp en termes de bits par pixel. Le gain maximal qui atteint 3.06 BPP est obtenu pour l'image 'Flower' dont la taille originale est de 240 KO. Le codage de Huffman contribue à la réduction de la taille de l'image compressée pour atteindre 216.72 K.O. A l'aide de la technique CHP, la taille de l'image compressée a été encore réduite à 124.88 K.O. En outre, le gain réalisé sur une variété d'images ('GrandPhare', 'Couple', 'Male', 'Flower', et 'Peppers') dépasse 2 Bpp. En général, le gain moyen est de 1.91 Bpp sur l'ensemble des images de test comme le montre le tableau (III.2).et figure (III.1).

Image	Codage de Huffman (CH)	Technique CHP	Gain
Cameraman	7.0448	5.0787	1.9661
Sailboat	7.2951	5.4116	1.8835
GrandPhare	7.0702	5.0758	1.9944
Airplane	5.6767	4.7564	0.9203
Flower	7.2241	4.1626	3.0615
Peppers	7.3975	4.9042	2.4933
Baboon	7.1673	6.1698	0.9975
Bikes	7.4055	5.9426	1.4629
Male	7.5486	5.3193	2.2293
Couple	7.2327	5.0986	2.1341
Débit moyen	7.1063	5.1920	1.9143

Tableau III. 2: Tableau comparatif des gains de débits (en Bpp) entre les techniques de codage CH et CHP

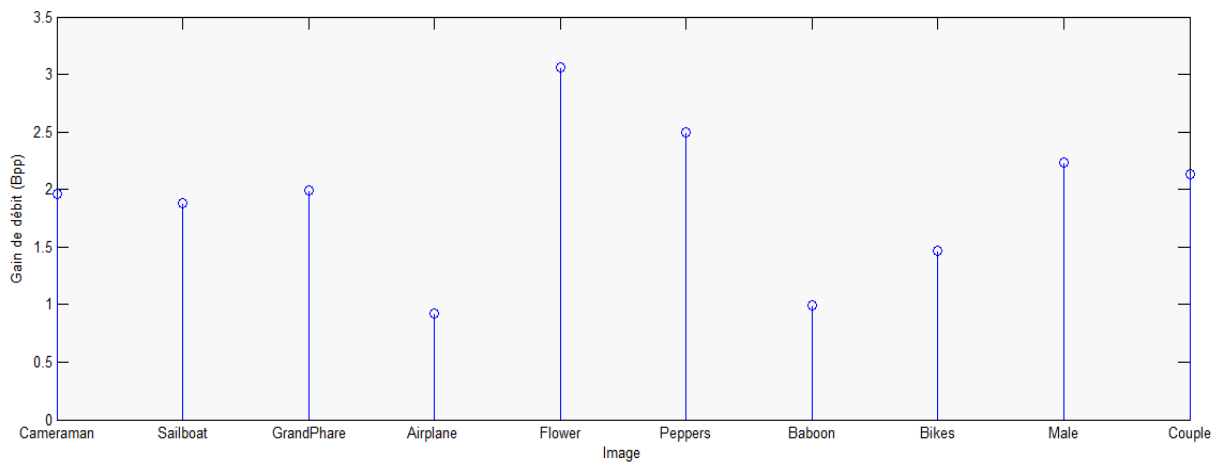


Figure III. 1: Gain en bits par pixel entre les techniques CH et CHP.

III.4.2 Evaluation de la performance des techniques de codage: CA (codage Arithmétique) et CAP (codage Arithmétique associé au codage prédictif)

Le protocole de compression à mettre en œuvre dans cette partie est comparable à celui de la partie précédente, sauf que l'on utilise le codeur Arithmétique au lieu du codeur de Huffman. Le tableau III.3 et la figure III.2 montrent les résultats obtenus.

		Codage Arithmétique (CA)			Codage Arithmétique associé au codage prédictif (CAP)		
Image	Taille de l'image originale	Entropie	Débit	Taille de l'image compressée	Entropie	Débit	Taille de l'image compressée
Cameraman	524288 Bits	7.0097	7.0100	459406 Bits	5.0551	5.0554	331309 Bits
	64 KO			56.0797 KO			40.4430 KO
Sailboat	2097152 Bits	7.2659	7.2660	1904731 Bits	5.3879	5.3879	1412415 Bits
	256 KO			232.511 KO			172.4139 KO
GrandPhare	3145728 Bits	7.0398	7.0399	2768194 Bits	5.0441	5.0442	1983444 Bits
	384 KO			337.9143 KO			242.119 KO
Airplane	8388606 Bits	5.6415	5.6415	5915515 Bits	4.7336	4.7336	4963525 Bits
	1024 KO			722.1088 KO			605.8990 KO
Flower	1966080 Bits	7.1913	7.1914	1767361 Bits	4.1402	4.1403	1017522 Bits
	240 KO			215.7423 KO			124.2092 KO
Peppers	2097152 Bits	7.3744	7.3744	1933164 Bits	4.8780	4.8781	1278765 Bits
	256 KO			235.9819 KO			156.0992 KO
Baboon	2097152 Bits	7.1391	7.1392	1871492 Bits	6.1471	6.1472	1611457 Bits
	256 KO			228.4536 KO			196.7111 KO
Bikes	3145728Bits	7.3720	7.3720	2898801 Bits	5.9007	5.9007	1466127 Bits
	384 KO			353.8575 KO			283.2356 KO
Male	8388608 Bits	7.5237	7.5238	7889231 Bits	5.2929	5.2929	5550037 Bits
	1024 KO			963.0409 KO			677.4948 KO
Couple	2097152Bits	7.2010	7.2011	1887721 Bits	5.0607	5.0608	1326647 Bits
	256 KO			230.4347 KO			161.9442 KO
Moyenne	/	7.0758	7.0759	/	5.1640	5.1641	/
Taux de compression moyen	/	1.1306	1.1305	/	1,5491	1,5491	/

Tableau III. 3: Tableau comparatif des performances de compression sans perte des techniques de codage: CA et CPA.

L'observation tirée de l'examen des résultats, permet clairement de voir des différences significatives, entre la méthode de codage Arithmétique appliquée toute seule au domaine spatial, et celle combinée avec le codage prédictif, pour l'ensemble des données du test.

Des résultats similaires à la première expérience ont été obtenus également en employant le codage arithmétique. Tout d'abord, la réduction significative de l'entropie est observée avec l'utilisation du codage prédictif, conduisant ainsi à une réduction appréciable du débit et de la taille des images compressées. Comme le montre le Tableau III.3, le codage prédictif combiné avec le codage arithmétique offre de faibles débits pour toutes les images par rapport au codage arithmétique appliqué absolument seul.

Comme on peut le constater au Tableau III.4 et figure III.2 les gains apportés par la méthode du codage Arithmétique appliquée conjointement avec le codage prédictive (CAP) sont importants et significatifs. Le gain obtenu varie entre 0.91 Bpp et 3.05 Bpp. Le gain s'élève à 3.05 Bpp pour l'image 'Flower', et dépasse 2 Bpp pour les images 'GrandPhare', 'Couple', 'Male', 'Flower', et 'Peppers. A titre d'exemple, le codage arithmétique permet une réduction de la taille qui atteint 215.74 KO. La technique CPA, quant à elle, conduit à une réduction supplémentaire de la taille qui passe à 124,21 KO. Globalement, le gain moyen avoisine 1.91 Bpp sur l'ensemble des images de test.

Image	Codage Arithmétique (CA)	Technique CAP	Gain
Cameraman	7.0100	5.0554	1.9546
Sailboat	7.2660	5.3879	1.8751
GrandPhare	7.0399	5.0442	1.9957
Airplane	5.6415	4.7336	0.9079
Flower	7.1914	4.1403	3.0511
Peppers	7.3744	4.8781	2.4963
Baboon	7.1392	6.1472	0.9920
Bikes	7.3720	5.9007	1.4713
Male	7.5238	5.2929	2.2309
Couple	7.2011	5.0608	2.1403
Débit moyen	7.0759	5.1641	1.9118

Tableau III. 4: Tableau comparatif des gains de débits (en Bpp) entre les techniques de codage CA et CAP.

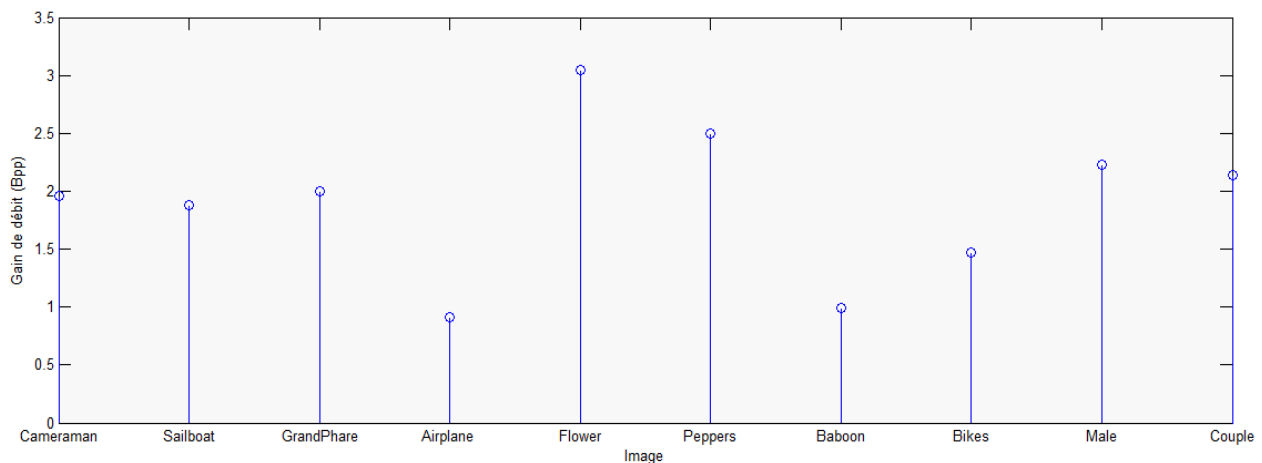


Figure III. 2: Gain en bits par pixel entre les techniques CA et CAP.

III.5 Evaluation de la performance en termes de complexité de calcul (temps de codage et décodage)

Les performances des techniques de codage: Huffman (CH), CHP, Arithmétique (CA), et CAP ont été évaluées en mesurant les temps de codage et décodage (en secondes) sur un ensemble de 10 images de taille différente et de caractéristiques distinctes. De plus, pour permettre une meilleure comparaison, nous avons utilisé la même machine PC pour effectuer les tests de mesures des temps de codage/décodage. Les résultats obtenus sont résumés dans les tableaux III.5 et III.6.

Comme on peut le voir dans le tableau III.5 et la figure III.3, la méthode de Huffman est très peu performante en termes de temps de calcul, et ce pour les phases de codage et décodage, par rapport à la méthode CHP qui permet des gains de temps considérables à la fois pour le codage et le décodage. En moyenne, le temps consacré au codage pour Huffman est de 155.97 secondes, alors que la méthode CHP ne nécessite que 80.43 secondes. Pour le décodage, Le temps moyen de Huffman paraît bien long: 2519.7 secondes en comparaison avec celui de la méthode CPH qui est significativement réduit à 758.51 secondes.

Pour la méthode du codage Arithmétique (CA), le temps moyen de codage est de 30.38 secondes, tandis que la méthode CAP ne prend que 23,86 secondes. Comme le montre le tableau III.6 et la figure III.4, durant le décodage, le temps moyen de la méthode CAP est légèrement inférieur (36.62 secondes) à celui de la méthode du codage Arithmétique appliquée de façon indépendante du codage prédictif (36.68 secondes).

Image	Codage de Huffman (CH)		Codage de Huffman associé au codage prédictif (CHP)	
	Temps de codage (s)	Temps de décodage (s)	Temps de codage (s)	Temps de décodage (s)
Cameraman	35.63830	305.8732	18.1744	123.3989
Sailboat	114.8270	1175	32.1953	338.6077
GrandPhare	186.7604	1934	90.0978	677.4337
Airplane	138.5661	3548.7	119.1098	1144.1
Flower	78.7634	2281	21.5822	176.4271
Peppers	125.8734	1122.7	56.2919	323.0142
Baboon	142.5986	1116.9	51.8660	473.0243
Bikes	158.7281	2018.7	106.3888	854.8633
Male	459.4922	10458	267.3313	3101.1
Couple	118.4281	1235.9	41.2720	373.0993
Temps moyen	155.9676	2519.7	80.4310	758.5068

Tableau III. 5: Comparaison des performances des deux techniques de codage CH et CHP en termes de temps de codage et décodage (en secondes).

Image	Codage Arithmétique (CA)		Codage Arithmétique associé au codage prédictif (CAP)	
	Temps de codage (s)	Temps de décodage (s)	Temps de codage (s)	Temps de décodage (s)
Cameraman	4.8025	5.7034	3.6044	5.7022
Sailboat	19.4174	23.4014	14.9484	23.1344
GrandPhare	28.7251	34.3934	21.7207	33.9632
Airplane	63.1080	78.9044	55.6777	77.6718
Flower	18.3986	21.6974	11.6009	21.6095
Peppers	19.3580	23.7274	14.2099	23.5206
Baboon	19.1187	22.9821	17.0929	23.0401
Bikes	29.8588	35.8610	24.6577	35.8764
Male	81.5998	96.7680	60.7712	98.2573
Couple	19.4300	23.3249	14.3203	23.4122
Temps moyen	30.3817	36.6763	23.8604	36.6188

Tableau III. 6: Comparaison des performances des deux techniques de codage CA et CAP en termes de temps de codage et décodage (en secondes).

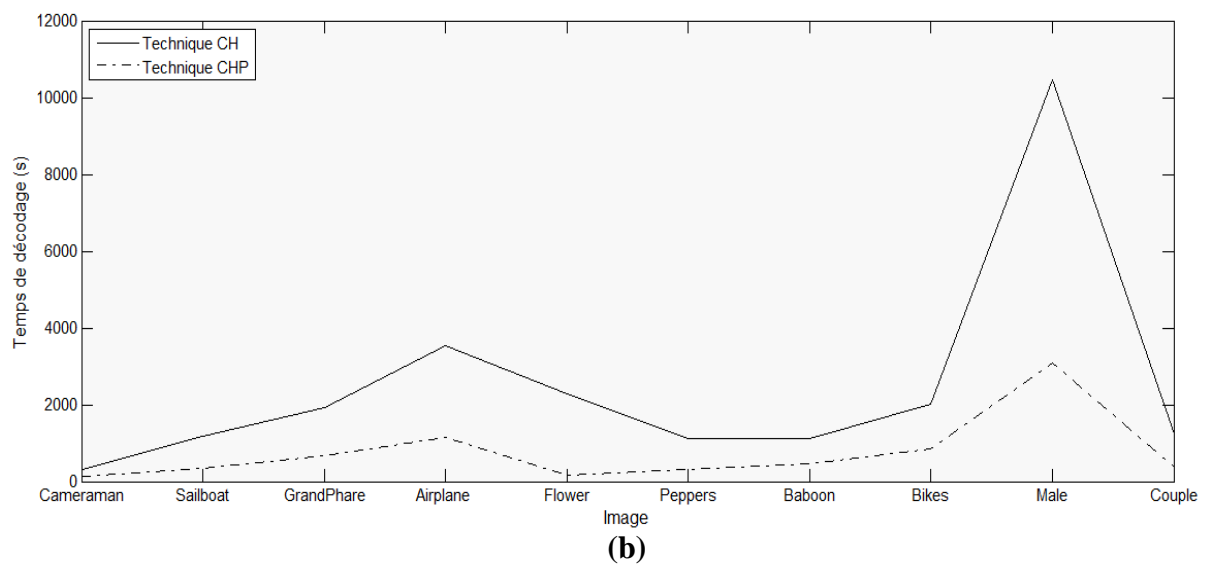
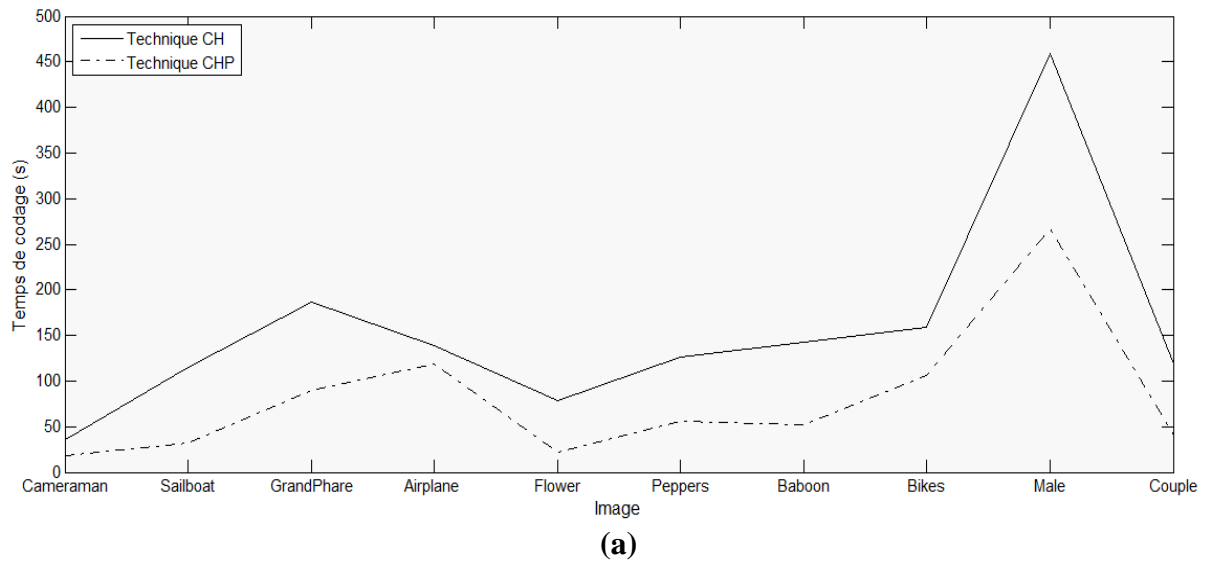


Figure III. 3: (a) Temps de codage (b) Temps de décodage en secondes entre les techniques CH et CHP.

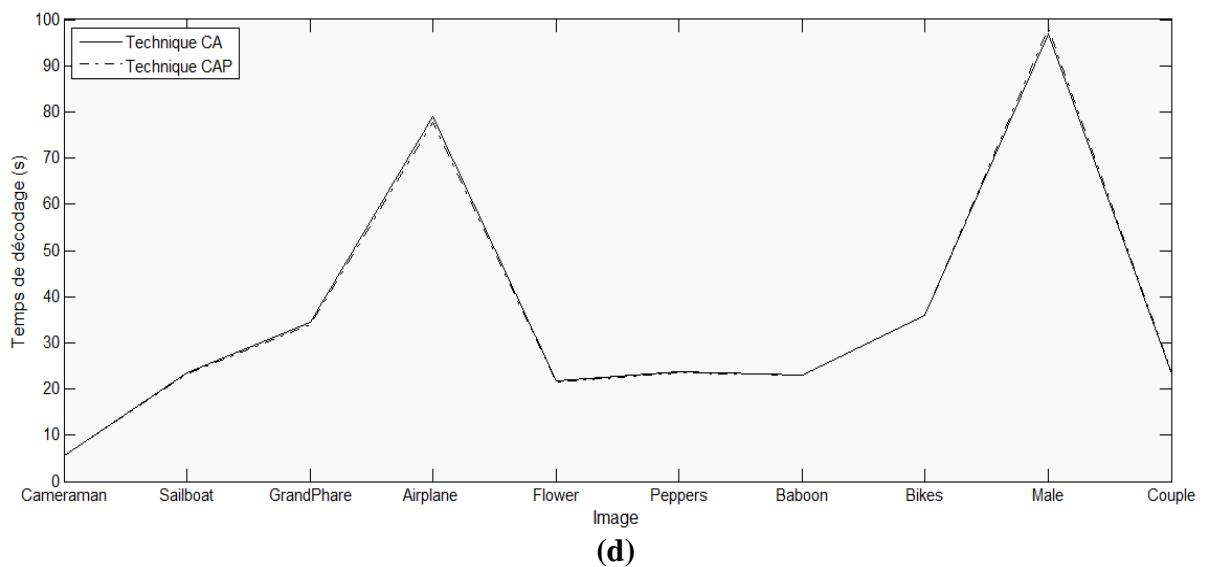
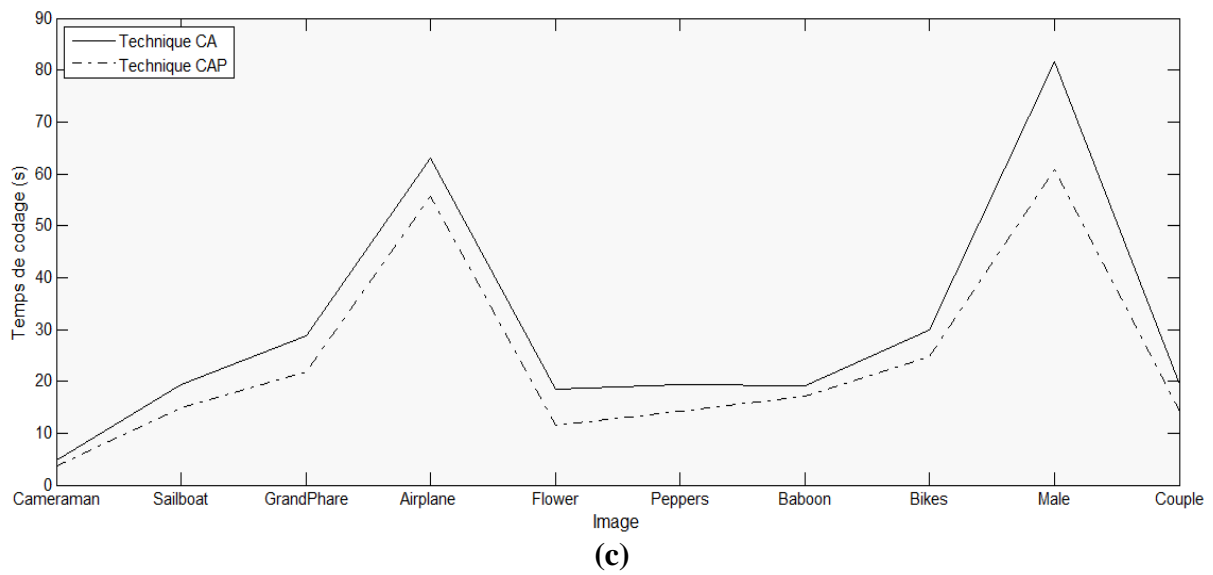


Figure III. 4: (c) Temps de codage (d) Temps de décodage en secondes entre les techniques CA et CAP.

III.6. Comparaison de la performance des techniques de codage utilisant le codage prédictif, les techniques CHP et CAP en termes de débit (en Bpp).

Nous présentons dans cette section les résultats des deux techniques de codage Huffman et Arithmétique appliquées conjointement avec le codage prédictif, en l'occurrence les techniques CHP et CAP, sur les mêmes données de test.

Image	Taille de l'image originale	Entropie	Technique CHP		Technique CAP	
			Débit	Taille de l'image compressée	Débit	Taille de l'image compressée
Cameraman	524288 Bits	5.0551	5.0787	332840 Bits	5.0554	331309 Bits
	64 KO			40.6299 KO		40.4430 KO
Sailboat512	2097152 Bits	5.3879	5.4116	1418623 Bits	5.3879	1412415 Bits
	256 KO			173.1718 KO		172.4139 KO
GrandPhare	2097152 Bits	5.0441	5.0758	1995905 Bits	5.0442	1983444 Bits
	256 KO			243.6407 KO		242.119 KO
Airplane	8388606 Bits	4.7336	4.7564	4987454 Bits	4.7336	4963525 Bits
	1024 KO			608.8201 KO		605.8990 KO
Flower	2097152 Bits	4.1402	4.1626	1023010 Bits	4.1403	1017522 Bits
	256 KO			124.8792 KO		124.2092 KO
Peppers	2097152 Bits	4.8780	4.9042	1285595 Bits	4.8781	1278765 Bits
	256 KO			156.9330 KO		156.0992 KO
Baboon	2097152 Bits	6.1471	6.1698	1617369 Bits	6.1472	1611457 Bits
	256 KO			197.4327 KO		196.7111 KO
Bikes	2097152Bits	5.9007	5.9426	2336743 Bits	5.9007	1466127 Bits
	256 KO			179.9675 KO		283.2356 KO
Male	8388608 Bits	5.2929	5.3193	5577700 Bits	5.2929	5550037 Bits
	1024 KO			680.8716 KO		677.4948 KO
Couple	2097152 Bits	5.0607	5.0986	1336580 Bits	5.0608	1326647 Bits
	256 KO			163.1567 KO		161.9442 KO
Débit moyen	/	5.1640	5.1920	/	5.1641	/
Taux de compression moyen	/	1,5491	1,5408	/	1,5491	/

Tableau III. 7: Tableau comparatif des performances de compression sans perte des techniques de codage: CPH et CPA.

Image	Technique CHP	Technique CAP	Gain
Cameraman	5.0787	5.0554	0.0233
Sailboat	5.4116	5.3879	0.0237
GrandPhare	5.0758	5.0442	0.0316
Airplane	4.7564	4.7336	0.0228
Flower	4.1626	4.1403	0.0223
Peppers	4.9042	4.8781	0.0261
Baboon	6.1698	6.1472	0.0226
Bikes	5.9426	5.9007	0.0419
Male	5.3193	5.2929	0.0264
Couple	5.0986	5.0608	0.0378
Débit moyen	5.1920	5.1641	0.0279

Tableau III. 8: Tableau comparatif des gains de débits (en Bpp) entre les techniques de codage CHP et CAP

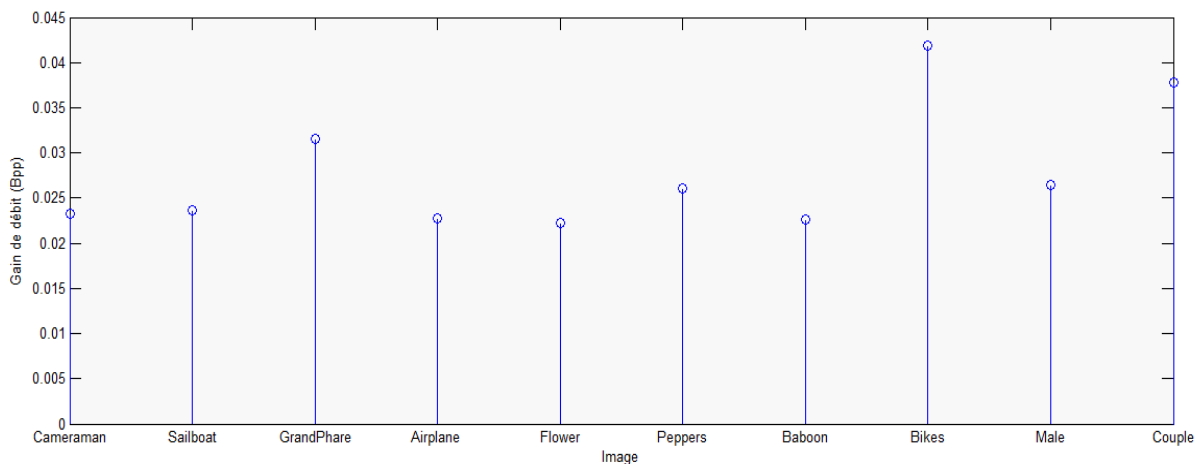


Figure III. 5: Gain en bits par pixel entre les techniques CHP et CAP.

A partir des résultats obtenus, récapitulés aux tableaux III.7, III.8 et figure III.3, nous constatons aussi que la variante CAP fournit de faibles débits pour toutes les images de test par rapport à la variante CHP. En général, le débit moyen diminue à 5.1641 bpp et avoisine celui de l'entropie qui vaut 5.1640 bpp. La technique CHP, quant à elle, offre un débit moyen de 5.1920 bpp. La même remarque s'applique pour l'analyse en termes du taux de compression. Notons qu'un codeur efficace tend à avoir une longueur moyenne, qui constitue le débit dans ce cas, très proche de l'entropie.

L'analyse des résultats obtenus met en évidence l'intérêt du codage prédictif appliqué en conjonction avec les techniques du codage source : Huffman et Arithmétique, pour améliorer la performance de compression sans perte. En effet, la redondance des pixels voisins est fortement réduite à travers l'extraction et le codage d'une nouvelle information dans chaque pixel. Cette nouvelle information, qui se définit comme la différence (ou l'erreur de prédiction) entre la valeur courante et celle prédite du même pixel, génère des valeurs plus faibles qui seront encodées, par un codeur de type longueur variable, de manière plus efficace que celle de l'application directe de l'encodeur sur l'image d'entrée.

L'analyse comparative des résultats qui a été menée sur plusieurs images de test montre également que la technique du codage CAP réalise la meilleure performance aussi bien en termes de débit que de temps de calcul

III.7. Conclusion

Ce chapitre s'est intéressé à l'analyse et l'évaluation des performances de compression sans pertes des techniques de codage source : Huffman et Arithmétique, appliquées de manière séparée et en association avec le codage prédictif, en l'occurrence les techniques CHP et CAP. Parmi les critères d'évaluation objective figurent le débit, le taux de compression et la taille de l'image compressée. La complexité de calcul a été explorée en mesurant le temps de codage et décodage tout en préservant le même environnement de travail. En utilisant un codage prédictif, dont le but est de réduire la corrélation existante entre pixels voisins, les performances de compression sans perte s'améliorent, avec des gains importants par rapport aux variantes ne comportant aucune étape de codage prédictif. Par ailleurs, notre étude met en évidence que de bien meilleurs résultats apparaissent lorsque le codage prédictif est combiné avec le codage Arithmétique, à la fois en termes de débit et de temps de calcul.

Conclusion générale

La compression est une technique dans laquelle un flux de séquence de données d'entrée est converti en un autre flux de séquence de données de plus petite taille. La plupart des flux d'entrée ont une redondance, l'objectif principal est donc de réduire cette redondance.

Dans ce mémoire, les performances de compression sans perte de deux techniques de codage de source: le codage de Huffman et le codage arithmétique, appliquées indépendamment et en conjonction avec le codage prédictif, ont été étudiées et comparées. Pour permettre une comparaison objective, les résultats de cette étude réalisée sur les quatre variantes présentées en l'occurrence CH (Codage de Huffman), CHP (codage de Huffman associé au codage prédictif), CA (Codage Arithmétique) et CAP (codage Arithmétique associé au codage prédictif), ont été analysés et évalués en termes de débit (en Bpp) et de complexité de calcul (en mesurant le temps de codage/décodage en secondes), en utilisant les mêmes données de test.

A la lumière des résultats présentés, le codage prédictif appliqué conjointement avec les techniques du codage source: Huffman et arithmétique, présente un grand intérêt pour améliorer la performance de compression sans perte. En effet, Les systèmes à base du codage prédictif tirent leur avantage de la réduction de la plage dynamique de la séquence de différence (ou l'erreur de prédiction). En effet, l'erreur de prédiction est généralement faible en raison de la corrélation spatiale des pixels voisins. Il en résulte que l'entropie de la séquence correspondant aux erreurs de prédiction sera faible et que, par conséquent, la séquence des erreurs de prédiction peut être codée en utilisant moins de bits.

Les résultats de notre étude, menée sur plusieurs images de test, montrent clairement que les techniques basées sur le codage prédictif (CPA et CPH), offrent les meilleures performances en termes de débit et de temps de calcul. Par ailleurs, des gains significatifs ont été enregistrés comparativement aux techniques appliquées de manière séparée (CH et CA).

Dans la perspective de développement de ce sujet, une étude de l'influence des coefficients des prédicteurs sur la performance de compression, portant sur un grand ensemble d'images aux caractéristiques bien distinctes, peut être réalisée.

Références Bibliographiques

- [1] K. Sayood. Introduction to data compression. Third Edition, Morgan Kaufmann Series, in Multimedia Information and Systems, 2017.
- [2] B. E. Usevitch. A tutorial on modern lossy wavelet image compression: Foundations of JPEG 2000. IEEE signal Processing Magazine, Vol. 18(5), pp. 22 –35, 2001.
- [3] John W. Woods. Multidimensional signal, image, and video processing and coding. Academic Press, Elsevier 2006.
- [4] D. Salomon. Data compression, the complete reference. Fourth edition, Springer 2007.
- [5] D. Salomon. A Concise Introduction to Data Compression. Springer 2008.
- [6] C. Gargour, M. Gabrea, V. Ramanchandran et J-M. Lina. “A short introduction to wavelets and their applications”. IEEE circuits and systems Magazine, Vol. 9 (2), pp. 57– 68, 2009.
- [7] M. Ghanbari. Standard Codecs: Image Compression to Advanced Video Coding. Third edition, Institution of Engineering and Technology, 2011.
- [8] Y.Q. Shi, H. Sun. Image and video compression for multimedia engineering. Fundamentals, algorithms, and standards. Third edition, CRC Press, 2019.
- [9] David R. Bull, Fan Zhang. Intelligent image and video compression: Communicating Pictures, Second Edition, Academic Press, Elsevier, 2021
- [10] Rafael C. Gonzalez & Richard E. Woods. Digital Image Processing. Fourth edition, Pearson 2018.
- [11] M. Sandeli, Traitement d'images par des approches bio-inspirées application à la segmentation d'images, Université Abdelhamid Mehri Constantine 2, 2014.
- [12] A.Medjaoui, F.Fares, Segmentation des images par contours actifs : application sur les Images Satellitaires à Haute Résolutions, Université Abou Bakr Belkaid, Tlemcen, 2012
- [13] T. Mohamed, D. Djafar, La découverte des concepts sémantiques cachés avec plusieurs niveaux d'abstraction pour la recherche d'images, Université kasdi Merbah, Ouargla, 2013

- [14] C. Houassine, Segmentation d'images par une approche biomimétique hybride. Université M'Hamed Bougara, Boumerdes, 2012.
- [15] M .T.Chikh, Amélioration des images par un modèle de réseau de neurones (comparaison avec les filtres de base), Université Abou Bakr Belkaid, Tlemcen, 2011
- [16] Y. Boudjit, Compression d'images sans perte par des techniques du codage source Mémoire de Master, Université Mohammed Seddik Ben Yahia-Jijel, Algérie, 2019.
- [17] B. Vincent. Développement de nouvelles techniques de compression de données sans perte. Mémoire de Maitres en Sciences, Science et Génie. Québec, 2008.
- [18] R. Sylvain, "La Compression de Données", Club Photoshop de Nantes, Conférence du 14 octobre 1999.
- [19] C. E. Shannon. "A mathematical theory of communication. Bell System Technical" Journal, 27 July October 1948.
- [20] M. BARNSLEY, Fractals Everywhere, Academic Press, Inc., 1988.
- [21] G. E. oien and S. Lepsoy. Fractal-based image coding with fast decoder convergence, Signal Processing, Vol. 40, pp. 105-117, October 1994.
- [22] G. Scorletti, "Traitement du Signal", Ecole d'ingénieur, Ecole Centrale de Lyon, 2013.
- [23] I.H. Witten, R. Neal, and J.G. Cleary. "Arithmetic Coding for Data Compression". Communications of the Association for Computing Machinery, Vol. 30, pp. 520–540, juin 1987.
- [24] Z. Djazia. Implémentation d'un environnement parallèle pour la compression d'images à l'aide des fractales, Thèse de doctorat, Batna, 2006.
- [25] B. Abraham, F. Gary. "A Mathematical Model For Estimating The Electiveness of Bigram Coding". Information Processing and Management, Vol. 12, pp. 111-116, 1976.

Résumé

Le traitement de l'image joue un rôle important dans le multimédia. Il permet de réduire la taille du fichier pour les besoins de stockage matériel et le temps de transition rapide. La technique de codage par prédiction est une méthode de compression d'image sans perte. Le codage par prédiction est le plus simple pour l'encodage et le décodage. Cela aide également à réduire les meilleures performances en termes de débit et la complexité de calcul. Le codage prédictif peut être amélioré à l'aide du le taux de compression et la taille de l'image compressée. Ce mémoire résume les algorithmes de codage de prédiction de base et leurs améliorations en utilisant diverses techniques.

Abstract

Image processing plays an important role in multimedia. It allows the file size to be reduced for hardware storage requirements and fast transition time. The predictive coding technique is a lossless image compression method. Prediction coding is the simplest for encoding and decoding. It also helps to reduce the best performance in terms of throughput and computational complexity. Predictive coding can be improved by using the compression ratio and the size of the compressed image. This paper summarises the basic predictive coding algorithms and their improvements using various techniques.

التلخيص

تلعب معالجة الصور دورًا مهمًا في الوسائط المتعددة. يساعد في تقليل حجم الملف لاحتياجات تخزين الأجهزة ووقت الانتقال السريع. تقنية تشفير التنبؤ هي طريقة ضغط صورة بدون فقد. ترميز التنبؤ هو الأسهل للترميز وفك التشفير. يساعد هذا أيضًا على تقليل الأداء الأفضل من حيث الإنتاجية والتعقيد الحسابي. يمكن تحسين الترميز التنبؤي باستخدام معدل الضغط وحجم الصورة المضغوطة. تلخص هذه الأطروحة خوارزميات ترميز التنبؤ الأساسية وتحسينها باستخدام تقنيات مختلفة.