

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

*Université Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj*

*Faculté des Sciences et de la technologie*

*Département d'Électronique*

# *Mémoire*

*Présenté pour obtenir*

**LE DIPLOME DE MASTER**

**FILIERE : ÉLECTRONIQUE**

**Spécialité : INDUSTRIES ÉLECTRONIQUES**

Par

- **Benneni Lamia**
- **Boussaad Wafa**

*Intitulé*

Étude et implémentation de techniques de RDH (Reversible Data Hiding) basées sur la méthode PVO (Pixel Value Ordering)

*Soutenu le : 26/06/2024*

*Devant le Jury composé de :*

<i>Nom &amp; Prénom</i>	<i>Grade</i>	<i>Qualité</i>	<i>Etablissement</i>
<i>Dr. Belguidoum Khaoula</i>	<i>MCB</i>	<i>Président</i>	<i>Univ-BBA</i>
<i>M. Abed Tarek</i>	<i>MAA</i>	<i>Encadreur</i>	<i>Univ-BBA</i>
<i>M. Djellal Djamel</i>	<i>MAA</i>	<i>Examineur</i>	<i>Univ-BBA</i>

*Année Universitaire 2023/2024*

# *Remerciements*

À l'issue de ce travail, nous adressons nos remerciements premièrement à ALLAH pour la volonté, le courage, la patience et la force qu'il nous a donné durant toute la période d'études.

On voudrait tout d'abord remercier sincèrement notre encadreur

***M. Abed Tarek***

non seulement pour son encadrement actif, mais aussi pour sa grande disponibilité, sa patience, ainsi que pour la générosité avec laquelle il a su partager ses connaissances et conseils.

Nous remerciant également tous les enseignants du

***DEPARTEMENT D'ELECTRONIQUE DE L'UNIVERSITÉ MOHAMED EL -  
BACHIR EL - IBRAHIMI DE BORDJ BOU ARRÉRIDJ***

plus spécialement les membres de jury de notre travail, et tous ceux qui ont collaboré de près ou de loin à la réussite de ce travail trouvent à la travers ces quelques lignes l'expression de notre profonde gratitude pour leur soutien et leur encouragement de tous les instants.

On vous en remercie.

# *Dédicaces*

*Tout d'abord, je tiens à remercier ALLAH*

*De m'avoir donné la force et le courage de mener à bien ce modeste travail.*

*Je tiens à dédier cet humble travail à :*

*A mon très cher père*

*Et particulièrement à très chère mère qui a toujours été là pour moi*

*A mon frère 'OUSSAMA', et mes sœurs 'AMINA, KARIMA' pour leur  
appui et leurs encouragements*

*A la mémoire de mon petit frère AKRAM, qu'Allah ait pitié de son âme*

*A ma chère binôme 'Wafa'*

*Enfin je dédie à tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de  
succès.*

*LAMIA BENNENI*

## *Dédicaces*

*Je remercie ALLAH, qui par la grâce de qui les bonnes actions sont accomplies.*

*Je dédie ce projet de fin d'études à tous ceux qui m'ont soutenu dans ma vie, notamment dans les moments de besoin :*

*A ma mère et mon père, que Dieu les préserve, une source d'encouragement et de soutien pendant vingt et un ans sans s'ennuyer.*

*A mon frère et ma petite sœur, le dos sur lequel je m'appuie.*

*A ma chère amie, WARDA, la main que je trouve dans des situations difficiles.*

*À ma plus chère amie universitaire LAMIA.*

*Et à mes chères amis " Leila, Soulef, Mayar et Ibtissem "*

*Et à tous les enseignants et professeurs qui ont contribué à mon développement et à mon éducation, ne serait-ce qu'un peu, au cours de ma carrière éducative.*

*Merci à tous.*

*Boussaad Wafa*

## Résumé :

Ce projet de fin d'études porte sur l'implémentation d'algorithmes d'intégration réversible de données sur des images en niveau de gris. Il s'agit de la méthode PVO et de méthodes issues de cette dernière qui sont les techniques IPVO et PPVO. La programmation en langage C++ des algorithmes de codage et de décodage est effectuée sous l'environnement C++Builder. Des tests de validité ainsi qu'une étude comparative des performances des trois techniques sont présentés dans ce mémoire.

**Mots-clés :** Intégration de données, stéganographie, RDH, PVO, programmation C++

## ملخص:

يتناول مشروع التخرج هذا تنفيذ خوارزميات تضمين البيانات القابل للعكس على الصور ذات التدرج الرمادي. ويشمل ذلك طريقة PVO والطرق المشتقة منها، والتي تتمثل في تقنيات IPVO و PPVO. تتم برمجة خوارزميات الترميز وفك التشفير بلغة C++ في بيئة C++Builder. يتم تقديم اختبارات الصلاحية بالإضافة إلى دراسة مقارنة لأداء التقنيات الثلاث في هذه الأطروحة.

**الكلمات المفتاحية:** تضمين البيانات، التخفي، RDH، PVO، برمجة C++

## Abstract:

This final year project is about the implementation of reversible data embedding algorithms on grayscale images. It involves the PVO method and methods derived from it, which are the IPVO and PPVO techniques. The programming of the encoding and decoding algorithms in C++ language is carried out in the C++Builder environment. Validity tests as well as a comparative study of the performance of the three techniques are presented in this thesis.

**Keywords:** Data embedding, steganography, RDH, PVO, C++ programming

# Table des matières

Introduction générale .....	1
<b>Chapitre I : Généralités sur les techniques d'intégration de données</b>	
I.1. Introduction.....	3
I.2. Définition de l'intégration de données .....	3
I.3. Les formes d'intégration de données .....	3
I.3.1 Intégration de données sur le signal audio .....	4
I.3.2. Intégration des données sur le signal vidéo .....	4
I.3.3. Intégration des données sur le signal d'image .....	4
I.4. Domaines d'application de l'intégration de données .....	4
I.4.1. Applications non-secrètes .....	4
I.4.2. Applications secrètes.....	5
I.4.2.1. Stéganographie.....	5
I.4.2.2. Tatouage numérique .....	5
I.5. Les types de technique d'intégration des données : .....	6
I.5.1. Intégration des données non-réversible .....	6
I.5.2. Intégration réversible de données (RDH).....	6
I.5.2.1. Principe .....	6
I.5.2.2. Domaines d'application .....	7
I.5.2.3. Performance de RDH .....	7
I.6. Conclusion .....	9
<b>Chapitre II : Techniques d'intégration réversible de données</b>	
II.1. Introduction.....	10
II.2. La méthode de l'expansion de la différence .....	10
II.2.1. Principe de fonctionnement.....	10
II.3. La méthode de l'expansion de l'erreur de prédiction (PEE) .....	12
II.3.1. Principe de fonctionnement.....	12
II.4. La technique « Pixel Value Ordering » PVO.....	13
II.5. Méthodes issues de la PVO .....	16
II.5.1. La méthode « Improved Pixel Value Ordering » IPVO.....	16
a. Principe de fonctionnement .....	17

b. Intégration des données.....	17
c. Extraction des données.....	18
II.5.2. La méthode « Pixel-based Pixel Value Ordering » PPVO.....	19
a. Principe de fonctionnement.....	19
b. Intégration des données.....	20
c. Extraction des données.....	21
II.6. Conclusion .....	22

### **Chapitre III : Implémentation des algorithmes de PVO, IPVO et PPVO sous C++ Builder**

III.1. Introduction.....	23
III.2. Présentation de l'environnement de développement.....	23
III.2.1. Environnement matériel .....	23
III.2.2. Environnement logiciel.....	23
III.2.2.1. C++ builder.....	23
III.2.2.2. Langage C++.....	24
III.3. Interface du programme d'intégration réversible de données .....	25
III.4. Implémentation des techniques de RDH .....	25
III.4.1. La technique PVO .....	26
III.4.1.1. Algorithme d'intégration de données .....	26
III.4.1.2. Algorithme d'extraction des données .....	29
III.4.2. La technique IPVO .....	32
III.4.2.1. Algorithme d'intégration de données .....	32
III.4.2.2. Algorithme d'extraction des données .....	33
III.4.3. La technique PPVO .....	33
III.4.3.1. Algorithme d'intégration de données .....	33
III.4.3.2. Algorithme d'extraction des données .....	36
III.5. Conclusion .....	38

### **Chapitre IV : Résultats et tests de performances**

IV.1. Introduction .....	39
IV.2. Test expérimental de l'application développée.....	39
IV.3. Comparaison des performances des méthodes implémentées .....	40
IV.3.1. Comparaison de la capacité d'intégration (EC) .....	41
IV.3.2. Comparaison des performances par le PSNR.....	42
IV.3.3. Graphes de comparaison des performances .....	44

IV.4. Conclusion.....	46
Conclusion générale.....	47
Références Bibliographiques.....	48



## Table des figures

Figure II.1. Prédiction pour le cas d'un un bloc d'une image 2x2 .....	12
Figure II.2. Forme de l'histogramme des différences .....	14
Figure II.3. Schéma de principe de fonctionnement de la technique PVO .....	15
Figure II.4. Exemple illustrant le fonctionnement de la technique PVO [20].....	16
Figure II.5. Représentation de l'histogramme des différences avant et après l'intégration .....	17
Figure II.6. Schéma du principe de fonctionnement de la technique PPVO .....	20
Figure.III.1. L'interface de C++ Builder .....	24
Figure III.2. Interface du programme d'intégration réversible de données.....	25
Figure III.4. Organigramme d'intégration des données par technique PVO .....	27
Figure III.3. Répartition des bits de l'entête .....	29
Figure III.5. Organigramme d'extraction des données par technique PVO.....	30
Figure III.6. Organigramme d'intégration des données par PPVO .....	34
Figure III.7. Organigramme d'extraction des données par technique PPVO .....	36
Figure IV.1. Exemple d'utilisation du programme d'intégration de données .....	39
Figure IV.2. L'extraction des données de Cameraman par PVO .....	40
Figure IV.3. Les images du test d'intégration de données .....	41
Figure IV.4. Comparaison des performances entre les méthodes PVO, IPVO et PPVO .....	45

## Table des tableaux

Tableau 1Tableau. IV. 1 : Capacité d'intégration maximale $EC_{max}$ .....	41
Tableau 2Tableau IV.2. Comparaison du PSNR (dB) pour $EC= 5000$ bits .....	42
Tableau 3Tableau IV.3. Comparaison du PSNR (dB) pour $EC= 10000$ bits .....	43
Tableau 4Tableau IV.3. Comparaison du PSNR (dB) pour $EC= 15000$ bits .....	43

## Table des abréviations

RDH	Reversible Data Hiding
PVO	Pixel Value Ordering
IPVO	Improved Pixel Value Ordering
PPVO	Pixel based Pixel Value Ordering
DE	Difference Expansion
PEE	Prediction Error Expansion
PSNR	Peak Signal-to-Noise Ratio
MSE	Mean Squared Error
EC	Embedding Capacity
bpp	bit per pixel
POSP	Proportion of Shifted Pixels
NCC	Normalized Cross Correlation Coefficient
LM	Location Map
CLM	Compressed Location Map

## Introduction générale

De nos jours, l'utilisation de documents, de supports ou de données numériques est devenue plus que courante. Que ce soit pour l'administratif, le professionnel et même le personnel, les médias numériques comptent une place prépondérante dans la vie de tous.

La révolution Internet, la disponibilité et l'efficacité des réseaux informatiques mondiaux pour la communication d'informations et de données numériques ont accéléré la popularité des médias numériques. Les images, vidéos et sons numériques ont été révolutionnés dans la manière dont ils peuvent être capturés, stockés, transmis et manipulés. Cela donne lieu à un large éventail d'applications dans l'éducation, le divertissement, les médias, la fabrication industrielle, la médecine et l'armée, entre autres domaines.

Avec ces développements, ont surgit des préoccupations et problèmes en matière de sécurité, de surveillance et d'utilisation de l'information par les utilisateurs. Par conséquent, des scientifiques, des chercheurs et des entreprises ont commencé à considérer ces préoccupations comme l'une de leurs principales priorités.

De ces préoccupations sont nées des techniques largement utilisées pour la sécurité et la protection des données, ce sont les techniques d'intégration de données. En fait, ces techniques ne sont pas tout-à-fait nouvelles, car elles ont existé avant l'ère du numérique par des filigranes sur papier, par exemple, utilisés pour indiquer de manière visible un éditeur particulier, pour décourager la contrefaçon de monnaie ou pour l'identification des droits d'auteur.

À l'ère moderne, la plupart des données et des informations étant stockées et communiquées sous forme numérique, la preuve de l'authenticité est effectuée grâce à des informations intégrées dans une image de manière à être imperceptibles pour les observateurs. Ainsi, il est possible d'identifier la source, le créateur, le propriétaire, le distributeur ou le consommateur autorisé d'un document ou d'une image.

L'intégration réversible des données (RDH) intègre des données secrètes dans une image hôte et peut récupérer parfaitement l'image hôte d'origine après que ces données secrètes ont été extraites de l'image marquée. La réversibilité de la RDH est très souhaitée dans certaines applications spéciales telles que le traitement d'images militaires et médicales, où aucune

distorsion n'est acceptable. De nombreuses méthodes RDH ont été proposées ces dernières années. Parmi elles, l'expansion des différences, le décalage d'histogramme et la technique dite PVO (Pixel Value Ordering). [1][4]

Dans ce projet, il s'agit d'une étude et implémentation de la technique PVO et deux autres techniques basées sur cette dernière méthode : IPVO et PPVO. PVO est une méthode basée sur la division de l'image en blocs distincts, les valeurs des pixels dans chaque bloc sont triées dans un ordre croissant. Les différences entre la valeur minimale et celle qui la suit, et entre la valeur maximale et celle qui précède sont exploitées pour éventuellement intégrer des bits de données. Les techniques IPVO et PPVO proposent des améliorations de l'algorithme de base de PVO.

Les programmes d'encodage de de décodage pour toutes ces techniques ont été écrits sous l'environnement de développement C++ Builder. Une série de tests ont permis de valider les algorithmes proposés et une étude comparative des performances a été menée afin de mettre en évidence la qualité et les avantages des différentes méthodes.

Le présent mémoire est divisé en quatre chapitres. Le premier propose une étude théorique concernant l'intégration des données, ses formes, ses types et ses applications.

Dans le deuxième chapitre, sont présentées quelques méthodes d'intégration de données, notamment la technique PVO et ses améliorations IPVO et PPVO. Le principe de fonctionnement de chaque méthode est expliqué.

Le troisième chapitre, permet d'expliquer et de mettre en œuvre les détails des algorithmes d'intégration et d'extraction pour chacune des trois méthodes.

Le dernier chapitre, est consacré à la présentation des résultats des tests afin de mettre en évidence les performances de toutes les méthodes, et de les comparer les unes aux autres.

**Chapitre I :**  
**Généralités sur les techniques d'intégration de**  
**données**

## **I.1. Introduction**

L'intégration de données est l'art de dissimuler un signal de message dans un signal hôte sans aucune distorsion perceptuelle du signal hôte. Le signal composite est généralement appelé signal stégo. Toute forme de communication repose sur un canal ou un support. Les communications de dissimulation de données, ou stéganographiques, reposent sur le canal utilisé pour transmettre le contenu hôte. Alors que le contenu stégo se déplace dans le monde entier, peut-être via Internet, ou par tout autre moyen habituellement déployé pour communiquer les signaux hôtes, le signal de message intégré et caché se déplace également.[1]

Dans ce chapitre, nous aborderons le principe de fusion ou d'intégration de données ainsi que les différentes techniques qui existent dans ce domaine.

## **I.2. Définition de l'intégration de données**

Les techniques d'intégration de données consistent à dissimuler ou intégrer un signal message dans un autre signal, comme un signal audio, des images, des vidéos...etc., afin que le message intégré devienne invisible tout en préservant les caractéristiques et la forme des signaux supports et messages. Cette opération est effectuée à des fins de sécurité ou de protection de la propriété des médias ou tout simplement pour les communications. [1]

La technologie d'intégration des données englobe de nombreuses méthodes et techniques qui permettent d'embarquer des données et de les extraire à la réception. Ces techniques peuvent être réversible ou irréversible, selon le fait que les données originales (support et message) sont récupérées sans ou bien avec pertes.

## **I.3. Les formes d'intégration de données**

Les formes de technique d'intégration des données dépendent du type de signal sur lequel les informations sont censées être transportées, ce sont :

### **I.3.1 Intégration de données sur le signal audio**

Les messages secrets peuvent être masqués dans des fichiers audio en modifiant légèrement certaines fréquences inaudibles à l'oreille humaine, ce qui consiste un processus de dissimulation réussi. [2]

### **I.3.2. Intégration des données sur le signal vidéo**

En raison de sa combinaison d'images et de son, la vidéo offre de nombreuses possibilités diverses pour masquer des informations confidentielles et constitue l'une des méthodes les plus difficiles pour l'intégration des données. [1]

### **I.3.3. Intégration des données sur le signal d'image**

La méthode la plus courante consiste à modifier légèrement les pixels d'une image numérique pour coder le message secret. Ces changements sont si minimes qu'ils sont invisibles à l'œil humain, mais ne peuvent être décodés qu'à l'aide d'un programme particulier.[3]

## **I.4. Domaines d'application de l'intégration de données**

Les domaines de la technologie d'intégration de données sont répartis selon l'utilisation, ou plutôt selon la finalité désirée, elle se divise en deux types d'applications :

### **I.4.1. Applications non-secrètes**

L'intégration des données ne se limite pas à la protection et la confidentialité, mais elle a un autre aspect, celui permettant de transporter avec un signal support des données additionnelles :

- Stockage de données : Le fait par exemple pour une entreprise de collecter des données d'un client pour les intégrer dans une image par exemple.
- Santé : Permet d'intégrer des données concernant un patient, issues d'analyses médicales, de rapports médicaux sur une image, une radiographie par exemple.
- Ressources humaines : La technologie d'intégration de données est utilisée dans le secteur des ressources humaines pour intégrer les données des employés.

## I.4.2. Applications secrètes

Dans la plupart des cas, l'intégration de données est utilisée comme processus de protection et de défense des informations, qu'elles soient personnelles ou fonctionnelles, par suite de l'émergence et du développement du vol numérique et du piratage. Nous aborderons ici deux domaines principaux qui apparaissent en abondance :

### I.4.2.1. Stéganographie

Le terme « stéganographie » est dérivé des mots grecs « steganos », qui signifie « couvert », et « graphein », qui signifie « écrire ». L'intention est de cacher des informations dans un support de telle manière que personne, à l'exception du destinataire prévu, ne connaisse son existence. [4] C'est l'art de dissimulation des messages et des données secrets, ou ce qu'on appelle l'art de la communication cachée, de sorte que même la connaissance de l'existence d'un message caché est considérée comme une affaire très secrète, c'est-à-dire que seul le destinataire prévu en connaît l'existence. [5] [6]

La stéganographie se présente sous trois formes principales (technique, linguistique et numérique). La stéganographie technique repose sur l'application de méthodes scientifiques pour cacher des messages secrets, tandis que la stéganographie linguistique exploite le langage naturel écrit à ces fins. Quant à la stéganographie numérique, elle est apparue avec l'avènement de l'informatique et utilise des fichiers électroniques ou des données multimédias numériques pour masquer des informations.

Les domaines d'application de la stéganographie sont divers, par exemple : [4] [5] [7]

- La sécurité ;
- Les communications secrètes ;
- Protection des droits d'auteur.

### I.4.2.2. Tatouage numérique

Le tatouage numérique représente une méthode ancienne utilisée pour prouver le droit d'auteur et indiquer des éditeurs spécifiques. Il s'agit d'un dessin ou d'un nom crypté intégré dans les documents afin de les protéger contre la fraude et l'usurpation d'identité. [1] [8]

Le tatouage numérique peut être utilisé dans plusieurs applications telles que : [9] [5]

- Surveillance de la diffusion

- Identification du propriétaire
- Preuve de propriété
- Suivi des transactions
- Authentification du contenu

## **I.5. Les types de technique d'intégration des données :**

### **I.5.1. Intégration des données non-réversible**

L'intégration non-réversible de données est une technique dans laquelle les informations cachées ne peuvent pas être récupérées à partir de données modifiées sans expertise ni accès aux données d'origine. Cela signifie qu'une fois les données modifiées pour contenir des informations cachées, les données originales ne peuvent pas être entièrement récupérées.

Les techniques d'intégration de données irréversibles incluent généralement les techniques suivantes :

- Insertion dans les bits les moins significatifs (LSB) : modification des bits les moins significatifs dans un support numérique (tel qu'un pixel dans une image) pour coder des données cachées.
- Tatouage numérique : intégration d'un tatouage ou d'un logo dans un contenu numérique pour suivre et protéger la propriété intellectuelle.
- Encodage perceptuel : exploiter les limites de la perception humaine pour intégrer des données dans des zones de médias difficilement visibles à l'œil ou à l'oreille humaine. L'intégration irréversible des données protège et transmet les informations sensibles sans attirer l'attention sur leur existence. Cependant, cela signifie également qu'une fois les informations cachées ajoutées, les données originales ne peuvent pas être entièrement récupérées.

### **I.5.2. Intégration réversible de données (RDH)**

#### **I.5.2.1. Principe**

L'intégration réversible de données RDH (Reversible Data Hiding) joue un rôle essentiel dans l'intégration de données confidentielles au sein des images. Cette technologie permet d'intégrer ces informations sans rien perdre, ce qui la rend particulièrement adaptée à des applications telles que le traitement d'images médicales ou militaires. [10]



### I.5.2.2. Domaines d'application

Les techniques d'intégration réversible de données sont utilisées dans de nombreux domaines qui sont sensibles aux informations, car les données sont masquées de manière réversible ne provoque pas de déformations permanentes du support de couverture. [11]

- L'intégration réversible des données d'imagerie médicale ;
- Communications secrètes (stéganographie) pour les communications secrètes dans l'armée ou les enquêtes criminelles ;
- Protection contre la fuite d'informations privées ;
- Tatouage numérique utilisé en intégrant un filigrane dans une image de couverture pour la protection de la propriété, l'authentification du contenu numérique, et la détection des contrefaçons et la protection des droits d'auteur.
- Protection des informations d'identification biométriques des individus ;
- Transmission d'informations satellite intégrées dans les images satellite garantissant la protection des données satellitaires contre tout accès et toute divulgation non autorisée.

### I.5.2.3. Performance de RDH

L'évaluation des performances des algorithmes d'intégration réversible de données se fait grâce à un certain nombre de mesures et paramètres parmi lesquels :

#### ▪ Capacité d'intégration (EC)

La taille maximale de la charge utile secrète pouvant être intégrée dans l'image de couverture est un élément clé. La capacité d'intégration est mesurée en termes de "nombre de bits" ou de "bits par pixel (bpp)" la valeur d'EC doit être suffisamment grande pour permettre l'intégration de l'ensemble des données secrètes. [11]

#### ▪ Proportion de pixels décalés (POSP)

Le POSP (Proportion of Shifted Pixels) est le rapport entre le nombre de pixels décalés et le nombre total de pixels dans l'image. Des valeurs élevées de POSP indiquent des niveaux élevés de distorsion dans l'image marquée. Le POSP est calculé à l'aide de l'équation (1), sa valeur du POSP devrait être minimum. [11]

$$\text{POSP} = \frac{N_s}{N_s + N_e} \quad (1)$$

- $N_s$  (Shifted pixels) est le nombre de pixels décalés
- $N_e$  (Expanded pixels) est le nombre de pixels expansés.

- **Erreur quadratique moyenne (MSE)**

L'erreur quadratique moyenne (MSE, Mean Squared Error) est utilisée pour comparer les valeurs des pixels entre l'image de couverture et l'image marquée. Cette métrique mesure la différence entre l'image originale et l'image finale après le processus de masquage des données. Elle est calculée à l'aide de l'équation (2) : [11]

$$MSE = \frac{1}{N \times M} \sum_{i=0}^M \sum_{j=1}^N [I(i, j) - I'(i, j)]^2 \quad (2)$$

Où  $I$  est l'image de couverture,  $I'$  est l'image marquée et  $M \times N$  est la taille des deux images. La gamme de MSE est  $[0, \infty [$ . La valeur de MSE entre l'image de couverture et le l'image marquée doit être minimale. [11]

- **Rapport signal/bruit de crête PSNR (Peak-Signal-to-Noise Ratio)**

Le PSNR est utilisé pour mesurer la qualité entre l'image de couverture  $I$  et l'image marquée  $I'$ . Le PSNR pour une image en niveaux de gris est calculé comme suit :

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \quad (3)$$

- **Indice de similarité structurelle (SSIM)**

Le SSIM est utilisé pour mesurer la similarité entre l'image de couverture et le texte marqué. Le SSIM est également une mesure de qualité pour le marqué. La plage du SSIM est  $[0, 1]$ . Pour les meilleures performances, sa valeur est très proche de 1.

Le SSIM peut être calculé comme suit :

$$SSIM = \left( \frac{2\mu_I \mu_{I'} + C_1}{\mu_I^2 + \mu_{I'}^2 + C_1} \right) \left( \frac{2\sigma_{II'} + C_2}{\sigma_I^2 + \sigma_{I'}^2 + C_2} \right) \quad (4)$$

$\mu_I$  est la moyenne de l'image de couverture

$\mu_{I'}$  est la moyenne de l'image marquée  $I'$

$\sigma_{II'}$  est la covariance de l'image de couverture et de l'image marquée  $\sigma_I^2$  et  $\sigma_{I'}^2$  sont respectivement les variances de l'image de couverture et de l'image marquée  $C_1$  et  $C_2$  sont les constantes de régularisation. [12]

- **Coefficient de corrélation normalisé (NCC)**

Le coefficient de corrélation normalisé vise à mesurer le degré de similitude entre l'image de couverture et l'image marquée. Il est calculé comme suit :

$$NNC = \frac{1}{N \times M} \sum_{i=0}^M \sum_{j=1}^N \frac{[I(i,j) - \text{mean}(I)] \times [I'(i,j) - \text{mean}(I')]}{\sqrt{\sigma_I^2 \sigma_{I'}^2}} \quad (5)$$

$M \times N$  est la taille de l'image de couverture  $I$

$I'$  est l'image marquée

$\sigma_I^2$  et  $\sigma_{I'}^2$  sont respectivement les variances de l'image de couverture et de l'image marquée.

[11]

## I.6. Conclusion

Dans ce chapitre, nous avons présenté une introduction aux techniques d'intégration de données. Nous avons abordé les différentes formes, types et domaines d'application de ces techniques. Nous avons également introduit les outils de mesure de performances des algorithmes d'intégration de données.

## **Chapitre II :**

# **Techniques d'intégration réversible de données**

## II.1. Introduction

L'intégration réversible de données vise à intégrer les informations dans l'image de manière invisible tout en assurant une extraction parfaite des données et une récupération complète de l'image l'originale.

L'intérêt pour cette technique RDH a conduit à la proposition de nombreuses méthodes et algorithmes, notamment l'expansion des différences (DE), l'expansion de l'erreur de prédiction (PEE) et le tri des valeurs de pixels (PVO). [13]

Ce chapitre est consacré à l'étude de ces techniques, particulièrement la technique dite PVO et ses algorithmes dérivés.

## II.2. La méthode de l'expansion de la différence

Les techniques basées sur l'expansion des différences (DE) ont beaucoup évolué au cours de la dernière décennie. Elles visent à intégrer de manière réversible des données dans une image hôte tout en réduisant la quantité d'informations auxiliaires à intégrer.[14]

### II.2.1. Principe de fonctionnement

L'expansion des différences (DE) est une technique d'intégration réversible de données. Cette technique permet d'intégrer des données sur la valeur de différence obtenue entre des paires de pixels. DE utilise des images en niveaux de gris de 8 bits constituées de paires de pixels  $(x, y)$  pour calculer la différence  $h$  et la moyenne  $l$  par l'équation 1 :

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor, \quad h = x - y \quad (1)$$

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor, \quad y = l - \left\lfloor \frac{h}{2} \right\rfloor \quad (2)$$

$x$  et  $y$  sont les valeurs de niveau de gris de deux pixels voisins satisfaisant :

$$0 \leq x, y \leq 255$$

$\lfloor x \rfloor$  Indique la fonction Plancher ou  $floor(x)$  qui permet d'arrondir au plus grand entier inférieur ou égal à  $x$ .

Des problèmes peuvent avoir lieu lorsqu'on effectue le processus d'insertion, ce sont les cas de débordement : overflow et underflow. Le premier est le cas où le résultat de la valeur modifiée du pixel, est supérieur à 255, tandis que le second cas, est le fait que le résultat de la

modification du pixel soit inférieur à 0. Pour éviter cela, les paires de pixels sur lesquelles il est possible d'intégrer des bits doivent satisfaire les conditions (3) et (4).

$$|h| \leq 2(255 - l) \text{ et } |h| \leq (2l + 1) \quad (3)$$

$$\begin{cases} |h| \leq 2(255 - l) & \text{si } 128 \leq l \leq 255 \\ |h| \leq (2l + 1) & \text{si } 0 \leq l \leq 127 \end{cases} \quad (4)$$

Ensuite, il est possible d'effectuer l'insertion d'un bit d'information  $b$  dans la valeur de différence  $h$ . On obtient la nouvelle valeur de différence  $h'$  après l'insertion du bit d'information  $b$  dans  $h$ , comme indiqué dans (5).

$$h' = 2h + b \quad (5)$$

$$b \in \{0, 1\}$$

Les nouvelles valeurs des pixels  $x'$  et  $y'$  sont calculées en utilisant (6)

$$x' = l + \left\lfloor \frac{h'+1}{2} \right\rfloor, \quad y' = l - \left\lfloor \frac{h'}{2} \right\rfloor \quad (6)$$

Afin de distinguer les paires de pixels utilisés pour l'intégration de celles non-utilisées, une carte de localisation doit être générée et intégrée avec la charge pure pour permettre l'extraction correctes des données. L'utilisation de cette carte de localisation présente l'inconvénient majeur de la méthode DE. Le processus d'extraction et de récupération présente des étapes très similaires à celles du processus d'intégration. Une des étapes clés consiste à extraire le bit de poids faible (LSB) de la différence calculée entre certaines paires de pixels modifiés. Cette valeur de différence obtenue permet ensuite de récupérer le bit d'information  $b$ . Les deux calculs réalisés dans ce cadre sont détaillés à l'étape (7)

$$b = LSB(h'), \text{ et } h = \left\lfloor \frac{h'}{2} \right\rfloor \quad (7)$$

Enfin, les valeurs originales de la paire de pixels  $(x, y)$  peuvent être obtenues en suivant les mêmes calculs que dans le processus d'intégration par l'équation (6).

En plus de la nécessité d'utiliser la carte de localisation, la méthode DE a posé plusieurs problèmes, plus particulièrement sa capacité d'intégration relativement faible qui peut atteindre au plus une capacité d'intégration théorique de 0.5bpp, avec une distorsion assez importante. D'autres algorithmes basés sur la méthode DE ont été développés afin d'augmenter la capacité d'intégration d'améliorer le niveau du PSNR, dans le but d'obtenir meilleure qualité d'image.

Parmi ces méthodes, la méthode dite de l'expansion de l'erreur de prédiction, a donné de très bonnes performances et a été également largement reprise par la suite.[15]

## II.3. La méthode de l'expansion de l'erreur de prédiction (PEE)

Cette technique vise à intégrer les données avec un impact visuel minimal et permet de récupérer l'image originale. Elle permet de corréliser les données avec les pixels voisins afin d'offrir une plus grande capacité d'intégration que la DE.

### II.3.1. Principe de fonctionnement

L'erreur de prédiction pour chaque pixel de couverture est déterminée à l'aide du pixel prédit. Ce pixel prédit est calculé par les pixels voisins. L'erreur de prédiction est expansée fournissant ainsi un espace libre pour intégrer les données.

Prenons l'exemple du pixel  $x$  dans une image en niveaux de gris, les pixels  $a, b, c$  dans la figure II.1 forment le contexte du pixel  $x$ . Ce contexte est utilisé afin de calculer la prédiction  $\hat{x}$  du pixel  $x$  en utilisant les règles de l'équation (8) [16]

$x$	$a$
$b$	$c$

Figure II.1. Prédiction pour le cas d'un bloc d'une image 2x2

$$\hat{x} = \begin{cases} \max(a, b) & \text{si } c \leq \min(a, b) \\ \min(a, b) & \text{si } c \geq \max(a, b) \\ a + b - c & \text{ailleurs} \end{cases} \quad (8)$$

Dans cette méthode, les pixels de la dernière colonne de la dernière ligne ne sont pas utilisés pour le processus d'intégration. [17]

L'erreur de prédiction est calculée comme :

$$p_{ee} = x_i - \hat{x}_i \quad (9)$$

Afin d'intégrer un bit  $i$  dans  $x$ , nous utilisons la représentation binaire de l'erreur de prédiction. Cette représentation est décalée vers la gauche d'un bit (équivalent à une multiplication par 2) afin de faire de l'espace pour recevoir un bit dans le LSB vacant. Si la représentation binaire  $p_{ee}$  est de longueur  $l$ , alors

$$\begin{cases} p_{ee} = b_{l-1}, b_{l-2}, b_0 \\ p'_{ee} = b_{l-1}, b_{l-2}, b_0 i = 2 \times p_{ee} + i \end{cases} \quad (10)$$

$p'_{ee}$  représente l'erreur de prédiction modifiée et  $i$  le bit à intégrer  $i \in \{0, 1\}$ .

$$x'_i = p'_{ee} + \hat{x}_i = x_i + p_{ee} + i \quad (11)$$

Au niveau du décodeur, l'extraction se fait en calculant d'abord la prédiction et l'erreur de prédiction  $p'_{ee}$ , le bit incorporé est extrait du bit de poids faible (LSB). La valeur du pixel d'origine est alors restaurée après avoir calculé l'erreur de prédiction initiale comme suit :

$$\begin{cases} P_{ee} = \left\lfloor \frac{P_{ee}'}{2} \right\rfloor \\ x_i = x'_i - P_{ee} - i \end{cases} \quad (12)$$

Pour la procédure d'extraction des bits et la récupération de la valeur originale, on procède de la façon suivante :

Calculer la prédiction d'erreur  $P_{ee}'$  et  $b$  en utilisant l'équation

$$\begin{cases} P_{ee}' = x_i - \hat{x}_i \\ b = \lfloor P_{ee}' \rfloor \text{mod}[2] \end{cases} \quad (13)$$

Une fois le processus précédent terminé, nous procédons à la récupération du support de couverture  $x$

$$\begin{cases} P_{ee} = \frac{P_{ee}' - b}{2} \\ x_i = \hat{x}_i + P_{ee} \end{cases} \quad (14)$$

## II.4. La technique « Pixel Value Ordering » PVO

La technique du tri des valeurs de pixels PVO est considérée parmi les meilleures techniques pour l'intégration réversible de données, car elle permet de garantir une qualité de l'image après le processus d'intégration. [5]

Tout cela est le résultat de sa dépendance au choix du pixel qui contient des informations, en plus d'une capacité d'intégration adaptée à l'intégration de données. Elle est considérée comme une technique très populaire très recherchée dans le domaine d'intégration réversible de données. [19]

Notre projet, étant axé principalement sur cette méthode, nous en discuterons en détail dans les paragraphes qui suivent.

### a. Principe de fonctionnement

Le principe de fonctionnement de la méthode du tri des valeurs des pixels est basé sur la comparaison des valeurs de  $n \times n$  pixels ( $2 \times 2$  dans notre cas) constituant un bloc de pixels.



Les pixels d'un bloc sont classés par ordre croissant ( $X_0, X_1, X_{n-1}, X_n$ ), chaque pixel gardera son emplacement dans l'image. Les différences, appelées erreurs de prédiction, entre chacun des deux pixels adjacents  $X_n - X_{n-1}$  et  $X_0 - X_1$  sont calculées. La figure suivante montre la forme de l'histogramme des erreurs de prédiction. Les différences égales à 1 ou -1 sont exploitées pour intégrer des données, toutes les autres devront être décalées. [20]

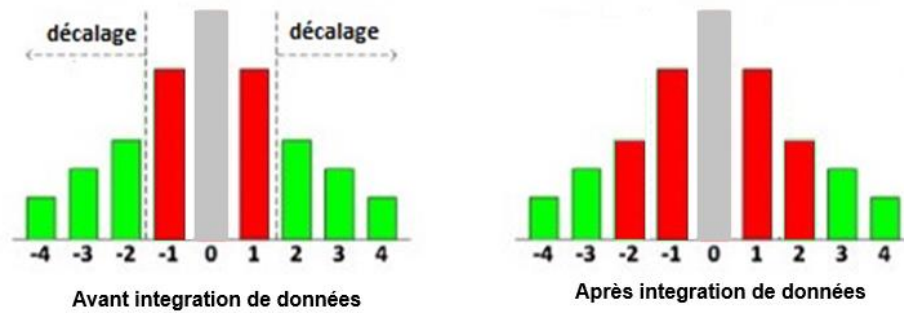


Figure II.2. Forme de l'histogramme des différences

### b. Intégration des données

Les blocs dont la différence  $X_n - X_{n-1}$  ( $PE_{max}$ ) est égale à 1 sont utilisés pour intégrer un bit  $b$ . Ce bit est ajouté au pixel ayant la plus grande valeur  $X_n$ , ceci n'affectera pas l'ordre des valeurs du bloc. De même, si la différence  $X_0 - X_1$  ( $PE_{min}$ ) est égale à -1, il est possible d'intégrer un bit  $b$ . Ce bit est soustrait du pixel ayant la plus petite valeur  $X_0$ , ceci n'affectera pas également l'ordre des valeurs du bloc. Il est évident, d'après cette méthode d'intégration, que la distorsion sera moindre du fait que la valeur d'un pixel n'est modifiée que de 1 au plus, l'image obtenue est ainsi de bonne qualité et est très similaire à l'image originale.

Afin qu'il n'y ait pas de confusion entre les blocs qui ont fait l'objet d'une intégration de données des autres, toutes les différences dont la valeur  $X_n - X_{n-1}$  est supérieure à 1 doivent être décalées en ajoutant 1 au pixel  $X_n$ . De même, toutes les différences dont la valeur  $X_0 - X_1$  est inférieure à -1 doivent être décalées en soustrayant 1 au pixel  $X_0$ .

Dans les équations (15) et (16) ( $PE'_{max}$ ) ( $PE'_{min}$ ) sont les valeurs modifiées des erreurs, les équations (17) et (18) indiquent l'intégration ou le décalage des valeurs  $X_n$  et  $X_0$  en fonction de la valeur de l'erreur de prédiction,  $X'_n$  et  $X'_0$  sont les valeurs modifiées des pixels  $X_n$  et  $X_0$ .

$$PE'_{max} = \begin{cases} PE_{max} & \text{si } PE_{max} = 0 \\ PE_{max} + b & \text{si } PE_{max} = 1 \\ PE_{max} + 1 & \text{si } PE_{max} > 1 \end{cases} \quad (15)$$

$$PE'min = \begin{cases} PEmin & \text{si } PEmin = 0 \\ PEmin - b & \text{si } PEmin = -1 \\ PEmin - 1 & \text{si } PEmin < -1 \end{cases} \quad (16)$$

$$X'_n = X_n + PE'max = \begin{cases} X_n & \text{si } PE'max = 0 \\ X_n + b & \text{si } PE'max = 1 \\ X_n + 1 & \text{si } PE'max > 1 \end{cases} \quad (17)$$

$$X'_0 = X_0 + PE'min = \begin{cases} X_0 & \text{si } PE'min = 0 \\ X_0 - b & \text{si } PE'min = -1 \\ X_0 - 1 & \text{si } PE'min < -1 \end{cases} \quad (18)$$

b est un bit de donnée à intégrer,  $b \in \{0, 1\}$

Il est important de noter que le fait d'intégrer des données ou de décaler des valeurs de différence, peut provoquer un débordement des valeurs de pixels (overflow ou underflow). Afin d'éviter cela, une carte de localisation doit être construite pour indiquer les blocs qui peuvent provoquer un débordement et les autres blocs utilisés pour l'intégration des données. Cette carte doit bien évidemment être intégrée avec la charge, mais sa taille est très réduite dans beaucoup de cas d'images. Cette méthode permet d'intégrer au plus 2 bits dans chaque bloc, dans les pixels ayant les valeurs minimales et maximales du bloc. [12] [20]

Le schéma de la figure suivante permet d'illustrer le fonctionnement de la technique PVO ainsi que le processus d'intégration.

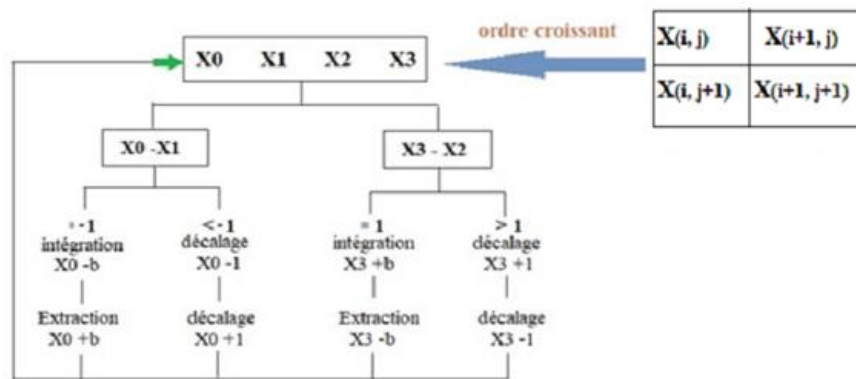


Figure II.3. Schéma de principe de fonctionnement de la technique PVO

### c. Extraction des données

Le processus d'extraction de données de son côté, utilise également le tri des valeurs des pixels d'un bloc. Cet ordre étant préservé lors du procédé d'encodage, et grâce à la carte de localisation il est possible de procéder à l'extraction des données et à la récupération de l'image

originale. Toutes les différences, indiquées par la carte de localisation comme utilisées pour l'intégration auront comme valeurs -2, -1, 1 et 2. Les différences -2 et -1 pour la différence des pixels de valeurs minimales, selon que l'on ait intégré un 1 ou un 0. Les différences 1 et 2 pour la différence des pixels de valeurs maximales, selon que l'on ait intégré un 0 ou un 1.

$$X_n = \begin{cases} X_n' & si & PE'max = 1, b = 0 \\ X_n' - 1 & si & PE'max = 2, b = 1 \\ X_n' - 1 & si & PE'max > 2 \end{cases} \quad (19)$$

$$X_0 = \begin{cases} X_0' & si & PE'min = -1, b = 0 \\ X_0' + 1 & si & PE'min = -2, b = 1 \\ X_0' + 1 & si & PE'min < -2 \end{cases} \quad (20)$$

La figure II.4 permet de montrer un exemple d'intégration et d'extraction par la méthode PVO. Des algorithmes détaillés des fonctions d'intégration et de décodage seront présentés dans la partie du chapitre 3 qui concerne l'implémentation de la méthode PVO.

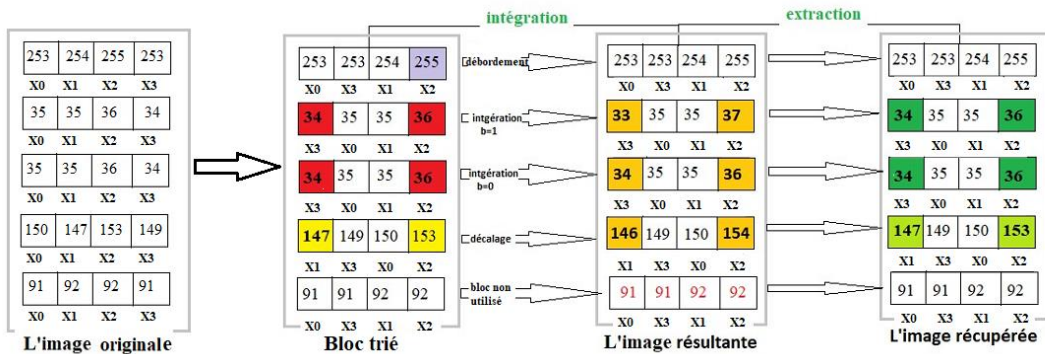


Figure II.4. Exemple illustrant le fonctionnement de la technique PVO [20]

## II.5. Méthodes issues de la PVO

La méthode PVO a été une technique très utilisée parmi les techniques de RDH. Plusieurs chercheurs ont proposé des techniques issues de la PVO. Nous introduirons dans ce qui suit deux d'entre elles qui sont les méthodes IPVO et PPVO, qui proposent toutes deux des améliorations de la méthode PVO originale. L'implémentation de ces dernières méthodes fait également partie des objectifs de ce projet.

### II.5.1. La méthode « Improved Pixel Value Ordering » IPVO

En raison de l'utilisation par la méthode PVO des erreurs de prédiction 1 et -1 et de la suppression de l'erreur de prédiction 0, les auteurs dans [18] ont proposé une amélioration de l'algorithme original, appelée la technique IPVO, qui vise à prendre avantage des blocs avec

l'erreur de prédiction 0 [18], et l'exploiter pour l'intégration des données, mais avec une idée différente, de sorte que IPVO fonctionne pour intégrer la plus grande quantité de données par rapport à la PVO et améliore donc la capacité d'intégration EC. [21]

L'implémentation détaillée des algorithmes d'encodage et de décodage IPVO sera expliquée au niveau du chapitre 3.

### a. Principe de fonctionnement

Le principe de IPVO est très similaire à celui de PVO dans la première partie consistant à diviser l'image en blocs et à les trier par ordre croissant. Dans cette méthode, deux erreurs de prédiction  $PE_{max}$  et  $PE_{min}$  sont calculées en prenant les différences  $X_n - X_{n-1}$  et  $X_0 - X_1$ .

En utilisant une méthode particulière du tri, qui sera développée dans la partie implémentation de ce mémoire, il est possible d'intégrer des données dans les cas où l'une ou l'autre des erreurs de prédiction  $PE_{max}$  ou  $PE_{min}$ , est égale à 1 ou égale 0. Ce qui accroît bien évidemment la capacité d'intégration.

Le schéma de la figure illustre de manière simple l'histogramme des différences (ou erreurs de prédiction) ainsi que le principe de fonctionnement de la technique IPVO.

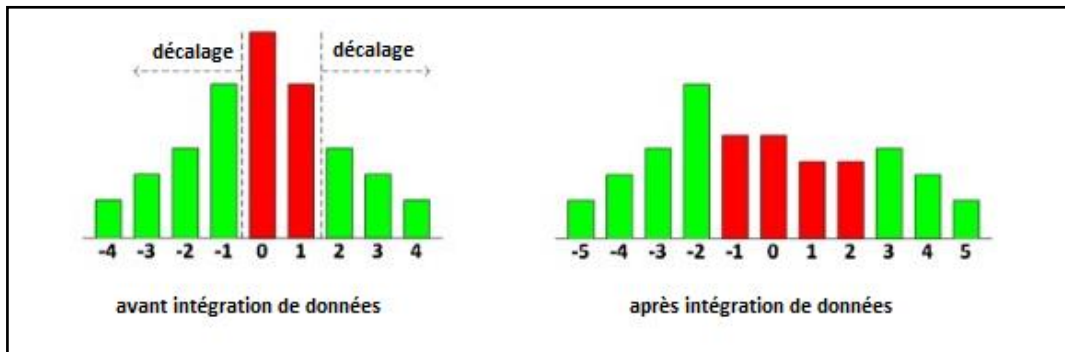


Figure II.5. Représentation de l'histogramme des différences avant et après l'intégration

### b. Intégration des données

Comme nous l'avons mentionné précédemment, le processus d'intégration de données dans IPVO est quelque peu similaire à PVO dans ses premières étapes, mais il en diffère par la manière d'intégrer les données. Les erreurs de prédiction  $PE_{max}$  et  $PE_{min}$  sont calculées par les équations suivantes :

$$PE_{max} = X_u - X_v \quad ; \quad \begin{cases} u = \min(\sigma(n), \sigma(n - 1)) \\ v = \max(\sigma(n), \sigma(n - 1)) \end{cases} \quad (21)$$

$$PEmin = X_r - X_s \quad ; \quad \begin{cases} r = \min (\sigma(0), \sigma(1)) \\ s = \max (\sigma(0), \sigma(1)) \end{cases} \quad (22)$$

Où  $\sigma(0) \dots \sigma(n)$  pour chaque pixel, indique l'index ou l'emplacement réel du pixel dans le bloc :

$$\begin{cases} \sigma(0) = \text{Emplacement du plus petit pixel du bloc} \\ \sigma(1) \\ \dots \\ \sigma(n-1) \\ \sigma(n) = \text{Emplacement du plus grand pixel du bloc} \end{cases}$$

Les valeurs de  $PEmax$  et  $PEmin$  permettront de déterminer les conditions d'intégration ou de déplacement des pixels  $X_n$  ou  $X_0$ . Les nouvelles valeurs des pixels  $X'_n$  et  $X'_0$  seront données par : [12]

$$X'_n = \begin{cases} X_n + b & \text{si } (PEmax = 1 \text{ ou } PEmax = 0) \\ X_n + 1 & \text{si } (PEmax > 1 \text{ ou } PEmax < 0) \end{cases} \quad (23)$$

$$X'_0 = \begin{cases} X_0 - b & \text{si } (PEmin = 1 \text{ ou } PEmin = 0) \\ X_0 - 1 & \text{si } (PEmin > 1 \text{ ou } PEmin < 0) \end{cases} \quad (24)$$

$b$  est un bit de donnée à intégrer,  $b \in \{0, 1\}$

### c. Extraction des données

Le processus de récupération dans IPVO n'est rien d'autre que le processus inverse d'intégration. Le calcul de  $PE'max$  et  $PE'min$ , qui constituent les erreurs de prédictions après le processus d'intégration, se fera de la même manière que celui de  $PEmax$  et  $PEmin$ . La relation entre  $PEmax$ ,  $PEmin$  et  $PE'max$ ,  $PE'min$  est donnée par : [11]

$$PE'max = \begin{cases} PEmax + b & \text{si } PEmax = 1 \\ PEmax + 1 & \text{si } PEmax > 1 \\ PEmax - b & \text{si } PEmax = 0 \\ PEmax - 1 & \text{si } PEmax < 0 \end{cases} \quad (25)$$

$$PE'min = \begin{cases} PEmin + b & \text{si } PEmin = 1 \\ PEmin + 1 & \text{si } PEmin > 1 \\ PEmin - b & \text{si } PEmin = 0 \\ PEmin - 1 & \text{si } PEmin < 0 \end{cases} \quad (26)$$

Les informations seront extraites de l'image résultante sur la base des relations suivantes, où  $X_n$  et  $X_0$  représentent les valeurs originales des pixels  $X'_n$  et  $X'_0$  :

$$X_n = \begin{cases} X_n' - 1 & \text{si } PE'max = -1, b = 1 \\ X_n' & \text{si } PE'max = 0, b = 0 \\ X_n' & \text{si } PE'max = 1, b = 0 \\ X_n' - 1 & \text{si } PE'max = 2, b = 1 \\ X_n' - 1 & \text{si } PE'max > 2 \text{ ou } PE'max < -1 \end{cases} \quad (27)$$

$$X_0 = \begin{cases} X_0' + 1 & \text{si } PE'min = -1, b = 1 \\ X_0' & \text{si } PE'min = 0, b = 0 \\ X_0' & \text{si } PE'min = 1, b = 0 \\ X_0' + 1 & \text{si } PE'min = 2, b = 1 \\ X_0' + 1 & \text{si } PE'min > 2 \text{ ou } PE'min < -1 \end{cases} \quad (28)$$

### II.5.2. La méthode « Pixel-based Pixel Value Ordering » PPVO

Les techniques PVO et IPVO et d'autres méthodes sont basées sur la division en blocs de l'image. Ainsi, deux pixels uniquement peuvent intégrer des données, il n'est donc possible d'intégrer que deux bits par bloc, les autres pixels sont ignorés [11]. Ceci implique une perte considérable en termes de capacité d'intégration. Pour pallier ce problème, une nouvelle technique de RDH a été proposée, c'est la technique PPVO. Cette technique, comme la PVO, repose sur le tri des valeurs d'un ensemble de pixels, mais cet ensemble de pixels ne constitue pas un bloc unique. La méthode PPVO considère chaque pixel de l'image individuellement en tant qu'élément capable d'intégrer des données, en fonction du reste de pixels de l'ensemble qui lui sont adjacents dans l'image [21], ce qui augmente considérablement la capacité d'intégration par rapport aux méthodes PVO et IPVO, tout en conservant la qualité de l'image.

#### a. Principe de fonctionnement

Son principe est différent de PVO, dans le sens où l'on divise l'image en blocs, mais on traite chaque pixel  $X$  dans le bloc comme un élément individuel, c'est-à-dire qu'on passe d'un pixel à l'autre dans l'ordre, et le reste des pixels environnants  $C$  dans le bloc constitue un contexte et est considéré comme un assistant dans le processus de prédiction et d'intégration.

Les valeurs contextuelles, la plus petite  $min(C)$  et la plus grande  $max(C)$ , sont déterminées. La valeur de la prédiction  $\hat{X}$  du pixel est calculée, ainsi que celle de l'erreur de prédiction  $PE = X - \hat{X}$  en fonction des valeurs de  $min(C)$  et  $max(C)$ .

La figure II.6 ci-après présente le principe de fonctionnement de la méthode PPVO. De plus amples détails concernant cet algorithme seront exposés dans la partie du chapitre 3 des implémentations. [21]

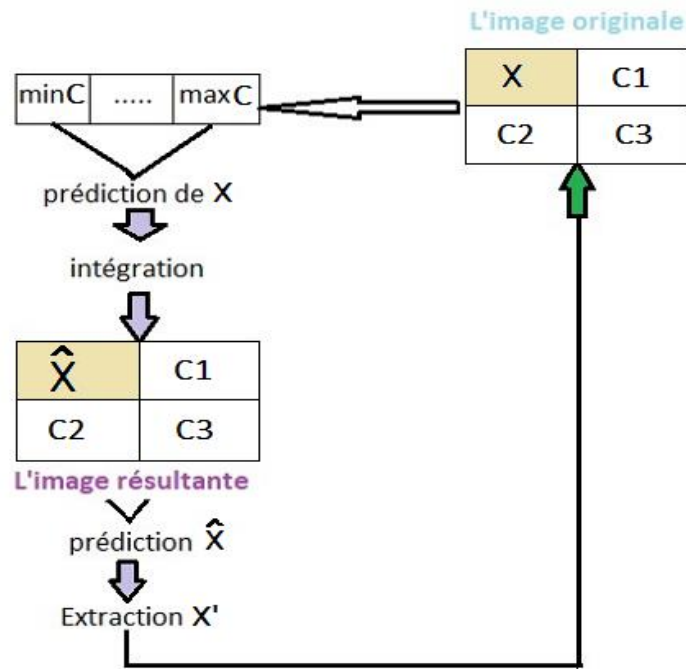


Figure II.6. Schéma du principe de fonctionnement de la technique PPVO

### b. Intégration des données

Le processus d'intégration des données dans PPVO, que nous avons précédemment abordé, consiste à prendre un point  $X$  ainsi que son contexte  $C$ . Les valeurs de  $\min(C)$  et  $\max(C)$  sont calculées, la prédiction  $\hat{X}$  du pixel  $X$  est évaluée de la manière suivante :

- Dans le cas où  $\min(C) \neq \max(C)$

$$\hat{X} = \begin{cases} \max(C) & \text{si } X \geq \max(C) \\ \min(C) & \text{si } X \leq \min(C) \end{cases} \quad (29)$$

- Dans le cas où  $\min(C) = \max(C)$

$$\hat{X} = \begin{cases} 254 & \text{si } \max(C) = \min(C) = VC = 254 \\ VC & \text{si } \max(C) = \min(C) = VC ; (1 \leq VC \leq 253) \end{cases} \quad (30)$$

L'erreur de prédiction est donnée par  $PE = X - \hat{X}$ . Le processus d'intégration des données exploite cette erreur de prédiction  $PE$  afin de déterminer le point  $X'$  qui est le pixel résultat de l'intégration.  $X'$  est alors calculé de la manière suivante : [11]

- Dans le cas où  $\min(C) \neq \max(C)$

$$X' = \begin{cases} X + b & \text{si } \hat{X} = \max(C) \text{ et } PE = 0 \\ X + 1 & \text{si } \hat{X} = \max(C) \text{ et } PE > 0 \\ X - b & \text{si } \hat{X} = \min(C) \text{ et } PE = 0 \\ X - 1 & \text{si } \hat{X} = \min(C) \text{ et } PE < 0 \end{cases} \quad (31)$$

- Dans le cas où  $\min(C) = \max(C)$

$$X' = \begin{cases} X + b & \text{si } VC = 254 \text{ et } PE = 0 \\ X - b & \text{si } VC \neq 254 \text{ et } PE = 0 \\ X - 1 & \text{si } VC \neq 254 \text{ et } PE < 0 \end{cases} \quad (32)$$

$b$  est un bit de donnée à intégrer,  $b \in \{0, 1\}$

### c. Extraction des données

Comme dans la fonction d'intégration, le processus d'extraction commence par évaluer la prédiction et l'erreur de prédiction  $PE'$  en fonction des valeurs  $\min(C)$  et  $\max(C)$  du contexte d'un pixel  $X$ . La même procédure de calcul est effectuée comme celle du processus d'intégration. [21]

L'extraction des données et la récupération de la valeur originale du pixel  $X$  se fait par les équations suivantes :

- Dans le cas où  $\min(C) \neq \max(C)$

$$X = \begin{cases} X' & \text{si } \hat{X} = \max(C) \text{ et } PE' = 0, b = 0 \\ X' - 1 & \text{si } \hat{X} = \max(C) \text{ et } PE' = 1, b = 1 \\ X' - 1 & \text{si } \hat{X} = \max(C) \text{ et } PE' > 1 \\ X' & \text{si } \hat{X} = \min(C) \text{ et } PE' = 0, b = 0 \\ X' + 1 & \text{si } \hat{X} = \min(C) \text{ et } PE' = -1, b = 1 \\ X' + 1 & \text{si } \hat{X} = \min(C) \text{ et } PE' < -1 \end{cases}$$

(33)

- Dans le cas où  $\min(C) = \max(C)$

$$X = \begin{cases} X' & \text{si } VC = 254 \text{ et } PE' = 0, b = 0 \\ X' - 1 & \text{si } VC = 254 \text{ et } PE' = 1, b = 1 \\ X' & \text{si } VC \neq 254 \text{ et } PE' = 0, b = 0 \\ X' + 1 & \text{si } VC \neq 254 \text{ et } PE' = -1, b = 1 \\ X' + 1 & \text{si } VC \neq 254 \text{ et } PE' < -1 \end{cases} \quad (34)$$



Ici  $X'$ ,  $\hat{X}$  et  $X$  sont respectivement les valeurs du pixel de l'image intégrée, la valeur de la prédiction dans cette image et la valeur du pixel restitué dans l'image décodée.

Il est à noter que le processus d'extraction doit se faire dans le sens inverse où a été faite l'intégration, en commençant par le dernier point ayant fait l'objet d'une intégration. Ce dernier point a son contexte de pixel inchangé, ce qui permet de récupérer de manière correcte sa valeur originale. Cette dernière doit être restituée avant de continuer le procédé d'extraction car elle fait partie du contexte du point précédent.

## II.6. Conclusion

Dans ce chapitre, nous avons brièvement discuté des techniques DE et PEE qui sont considérées comme le point de départ pour beaucoup de méthodes appartenant au même axe de développement dont la méthode PVO.

Nous avons fait le point sur la méthode PVO, ainsi que les techniques qui constituent des améliorations pour cette méthode.

Nous avons développé d'une manière un peu plus détaillée les principes de fonctionnement, les procédés d'intégration et d'extraction des techniques PVO, IPVO et PPVO qui feront l'objet d'un exposé détaillé, dans le chapitre suivant, de leurs algorithmes et leurs implémentations.

**Chapitre III :**  
**Implémentation des algorithmes de PVO, IPVO**  
**et PPVO sous C++ Builder**

## III.1. Introduction

Dans toute science, étudier quelque chose nécessite de l'appliquer de manière réaliste et d'expérimenter pour découvrir la validité de l'étude. Cette expérimentation peut être implémentée de plusieurs manières différentes, notamment à travers la programmation.

Dans ce chapitre, nous discuterons de l'implémentation des techniques PVO que nous avons étudiées précédemment, en utilisant le langage C++. Cette implémentation permettra de valider les algorithmes étudiés ainsi que d'expérimenter et de comparer leurs performances.

## III.2. Présentation de l'environnement de développement

Dans cette section, nous présentons l'environnement logiciel (software) et matériel (hardware) qui ont été utilisés pour la partie pratique de ce projet.

### III.2.1. Environnement matériel

La programmation, dans ce travail, a été réalisée en utilisant un PC (Laptop DELL) dont les caractéristiques :

- Processeur : Intel(R) Core (TM) i3-6100U CPU @ 2.30GHz 2.30 GHz
- Mémoire installée (RAM) : 4,00 Go
- Type du système : Système d'exploitation 64 bits, processeur x64
- Carte graphique : Intel (R) HD Graphics 520

### III.2.2. Environnement logiciel

Nous utilisons pour les implémentations, l'environnement « Embarcadero C++ Builder 11 – Community Edition ».

#### III.2.2.1. C++ Builder

C++ Builder est un environnement de développement intégré (IDE) complet permettant de créer des applications sous Windows. Il comprend un éditeur de code C++ ou C. En plus des modèles, il fournit également un accès intégré aux principales bases de données locales et Bluetooth, ainsi qu'aux fonctionnalités l'internet de chose, ainsi qu'une interface utilisateur visuelle au pixel près. [22]

L'interface C++ Builder est organisée en plusieurs zones principales qui permettent la création et le développement d'applications C++ [23]. La figure suivante montre les zones les plus importantes pour le développement d'applications sous l'environnement C++ builder :

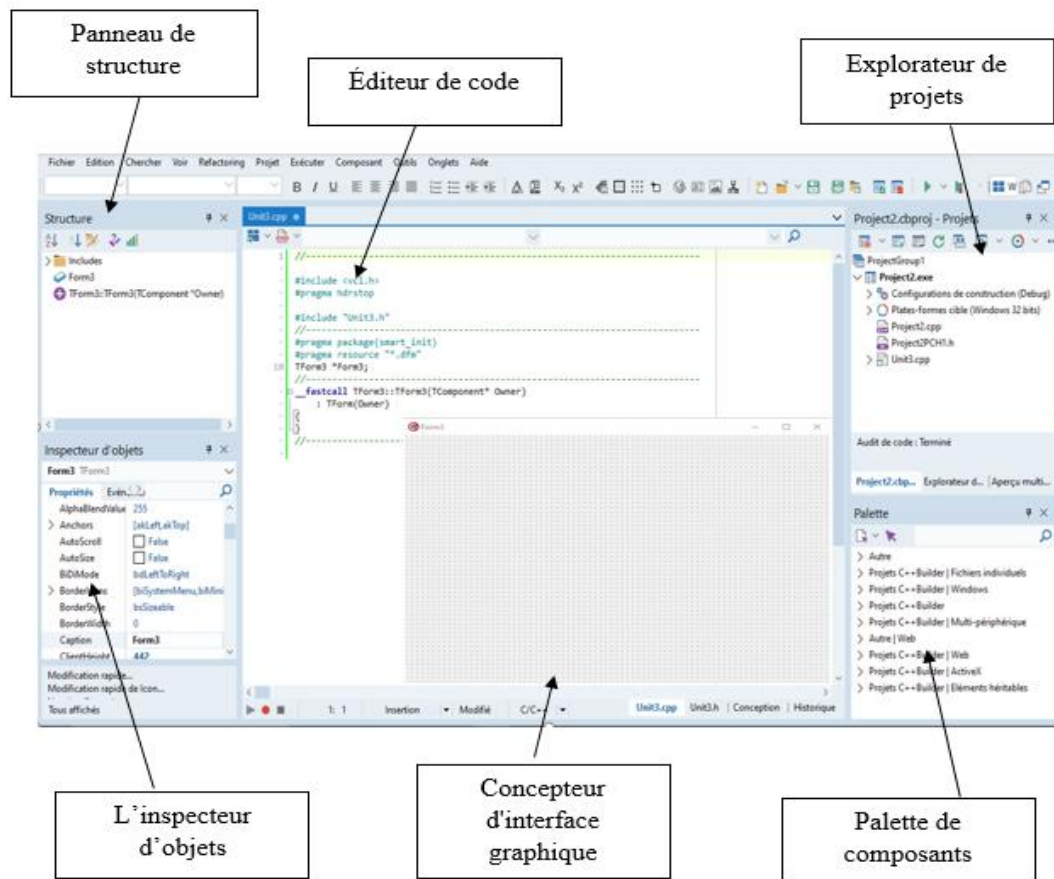


Figure.III.1. L'interface de C++ Builder

### III.2.2.2. Langage C++

Le langage de programmation le plus couramment utilisé par les développeurs, notamment pour les applications, est le C++, qui est un langage de programmation orienté objet.

Le langage C++ contient une bibliothèque standard, comprenant de nombreux types de classes et de fonctions qui permettent de mettre en œuvre les structures de données les plus importantes (vecteurs dynamiques, listes chaînées, etc.) et les algorithmes les plus courants. Parmi les environnements de développement nous citons : Anjuta DevStudio, C++ Builder, Code Blocs (open source), Dev-C++, Eclipse (open source), Microsoft Visual C++.

### III.3. Interface du programme d'intégration réversible de données

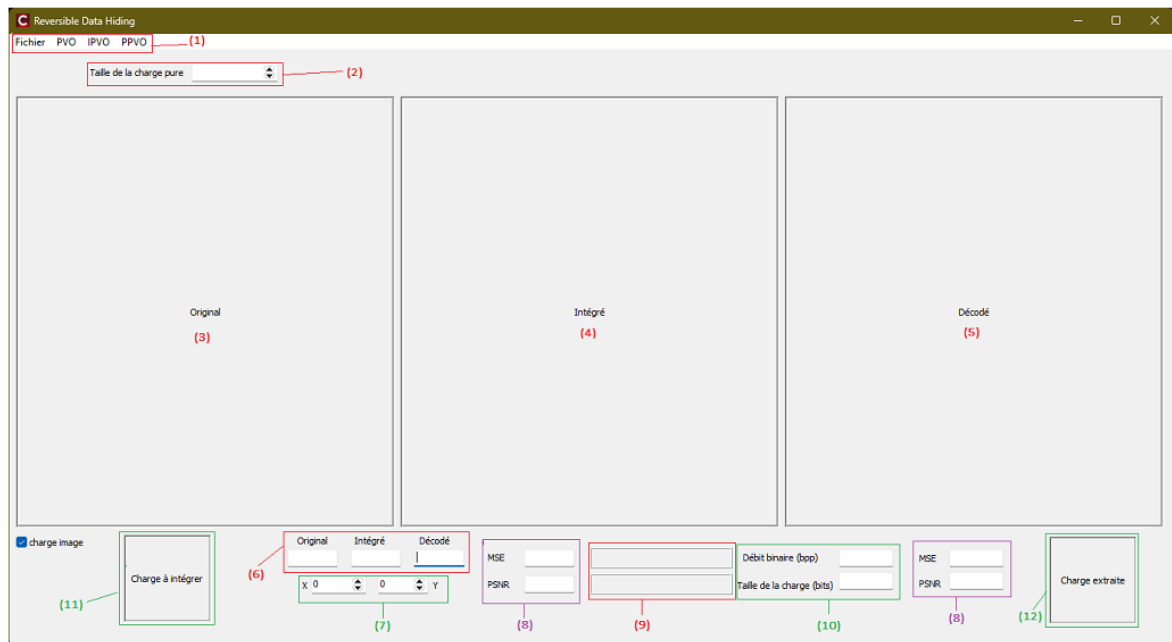


Figure III.2. Interface du programme d'intégration réversible de données

1. Menu principal
2. Introduction de la taille de données à intégrer
3. L'image originale
4. L'image résultante
5. L'image décodée
6. Affichage des valeurs de niveau de gris des pixels
7. Affichage des coordonnées du pixel indiqué en (6)
8. Valeurs de MSE et PSNR après le codage et après le décodage
9. Messages après décodage
10. Taux d'intégration en bits par pixel
11. Charge à intégrer (sous forme d'image)
12. Charge extraite (sous forme d'image)

### III.4. Implémentation des techniques de RDH

Dans cette partie, nous allons exposer l'implémentation de l'ensemble des algorithmes d'intégration et de récupération de données que nous avons étudié dans le chapitre précédent.

### **III.4.1. La technique PVO**

Dans le deuxième chapitre, nous avons étudié toute la théorie qui entoure la technique PVO. Dans ce qui suit nous nous intéresserons à l'implémentation détaillée de cet algorithme.

#### **III.4.1.1. Algorithme d'intégration de données**

Le schéma de la figure III.4 illustre l'organigramme de la fonction d'intégration de données par la technique PVO. [20]

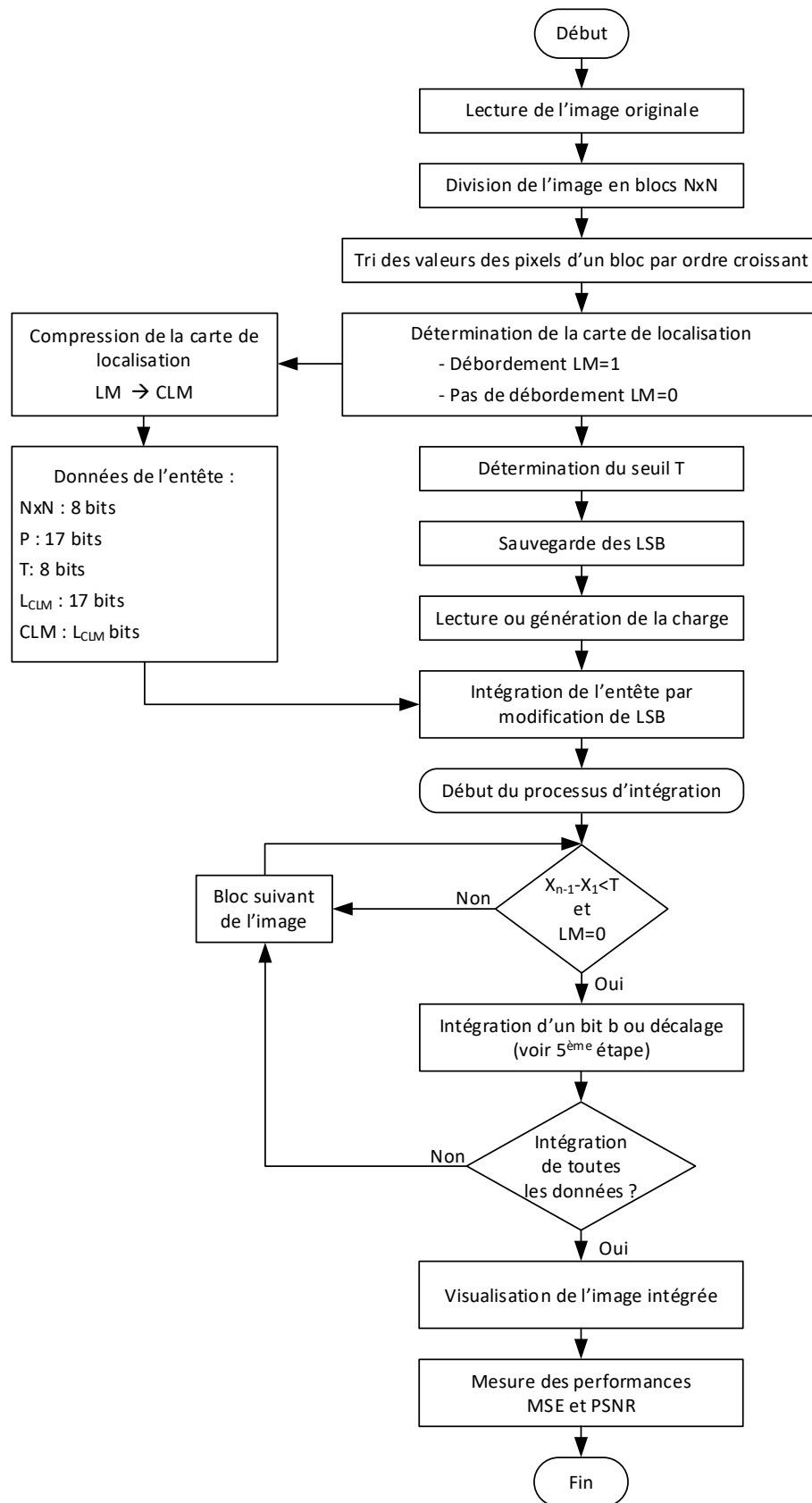


Figure III.4. Organigramme d'intégration des données par technique PVO

- Première étape : Division de l'image en blocs et tri des valeurs

Diviser l'image en blocs NxN (2x2 dans notre application) et trier les valeurs de pixels dans chaque bloc par ordre croissant :  $\{X_0, X_1, \dots, X_{n-1}, X_n\}$  .

- Deuxième étape : Génération de la carte de localisation LM

La carte LM permet de distinguer les blocs qui peuvent intégrer des données et les blocs dans lesquels un débordement peut se produire et dans lesquels on ne peut pas intégrer de données. La carte est construite de la manière suivante :

LM=1 pour indiquer la possibilité de débordement, ce qui signifie que le bloc doit être ignoré. Ceci dans les conditions suivantes

$$\begin{cases} X_n = 255 & \text{et} & X_n - X_{n-1} \geq 1 \\ X_0 = 0 & \text{et} & X_0 - X_1 \leq -1 \end{cases}$$

LM=0 indique que le bloc est capable d'inclure des données ou un décalage de valeur

La carte de localisation est compressée en utilisant une fonction de compression de type « Run Length ».

- Troisième étape : Sélection des blocs pour l'intégration

Dans cette étape, il s'agit de déterminer un seuil pour l'intégration. Ce seuil permet de choisir les blocs qui vont recevoir les données. La valeur du seuil est initialisée à 1 puis incrémenter à chaque fois en comptant le nombre de blocs capables d'intégrer des données, ces blocs répondent aux conditions :

$$LM=0 \quad \text{et} \quad X_{n-1} - X_1 < T$$

Le processus itératif s'arrête lorsque les nombres de blocs qui peuvent recevoir des données est suffisant pour la charge (payload) à intégrer.

La condition sur le seuil permet d'assurer une intégration de données dans les zones lisses de l'image.

- Quatrième étape : Insertion des données d'en-tête

Avant de procéder à cette insertion, les LSB des pixels qui doivent recevoir l'entête sont sauvegardés pour être insérés avec la charge. Les données de l'entête sont insérées dans l'image par modification du LSB des premiers pixels de l'image. Elles permettent de passer



les informations nécessaires au processus de décodage ultérieur. Ces données dont la taille totale en bits est  $52+L_{CLM}$  sont :

- La taille du bloc : 8 bits ;
- La valeur du seuil T : 8 bits ;
- La taille de la charge P : 17 bits ( $2^{17}$  est la charge maximale si tous les blocs peuvent intégrer 2 bits pour une image 512x512).
- La taille de la carte compressée  $L_{CLM}$  : 17 bits ;
- La carte compressée CLM :  $L_{CLM}$  bits



Figure III.3. Répartition des bits de l'entête

➤ Cinquième étape : Insertion des données

Afin de pouvoir vérifier la validité des algorithmes, une charge aléatoire est générée. Il est également possible d'utiliser une image binaire ou une quelconque chaîne de bits.

Le processus d'insertion des données commence après le dernier pixel d'insertion de l'entête. L'intégration qui consiste à effectuer des modifications sur les pixels  $X_0$  et  $X_n$ , de la manière suivante :

$$X'_n = \begin{cases} X_n + b & \text{si } X_n - X_{n-1} = 1 \\ X_n + 1 & \text{si } X_n - X_{n-1} > 1 \\ X_n & \text{si } X_n - X_{n-1} = 0 \end{cases}$$

$$X'_0 = \begin{cases} X_0 - b & \text{si } X_0 - X_1 = -1 \\ X_0 - 1 & \text{si } X_0 - X_1 < -1 \\ X_0 & \text{si } X_0 - X_1 = 0 \end{cases}$$

$X'_0$  et  $X'_n$  sont les pixels résultant de l'intégration, les bits  $b$  sont les bits de données de la charge à intégrer,  $b \in \{0, 1\}$

➤ Sixième étape : Mesure des performances MSE et PSNR

### III.4.1.2. Algorithme d'extraction des données

La figure III.5 montre l'organigramme d'extraction de données de la méthode PVO permettant la récupération de l'image originale et de la charge. [20]

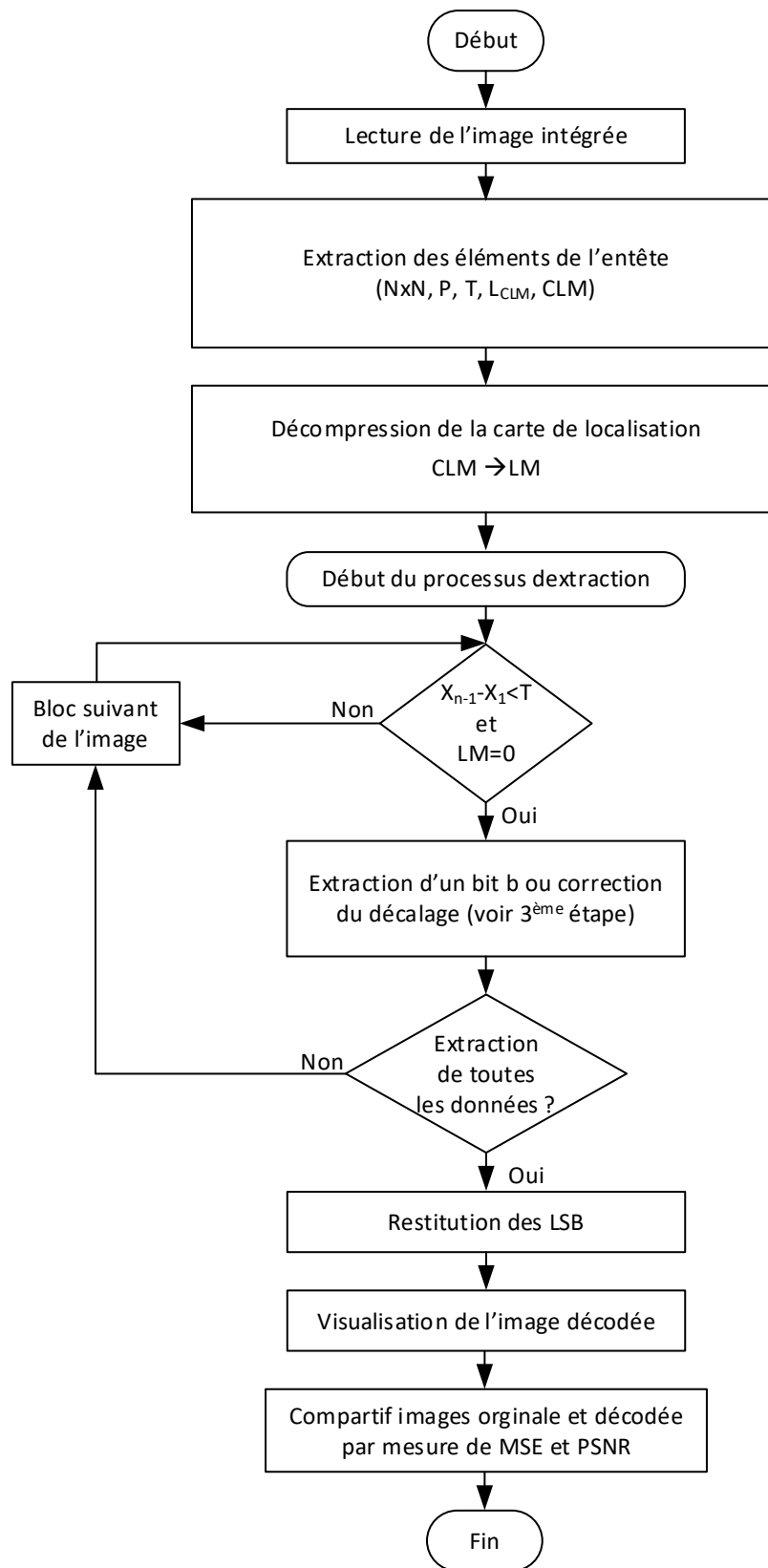


Figure III.5. Organigramme d'extraction des données par technique PVO

➤ Première étape : Extraction de l'entête

Récupérer les LSB de tous les premiers pixels utilisés pour véhiculer les éléments de l'entête et de les convertir en décimal. Les données extraites sont : la taille du bloc, la valeur d seuil, la taille P de la charge, la taille  $L_{CLM}$  de la carte compressée, et la carte de localisation compressée CLM.

➤ Deuxième étape : Décompression de la carte de localisation CLM

CLM est décompressée à l'aide d'une fonction de décompression « Run Length », la carte LM est donc restituée.

➤ Troisième étape : Extraction de la charge et des LSB sauvegardés

Diviser l'image résultante en blocs de taille  $N \times N$  ( $2 \times 2$  dans l'application) et trier les valeurs de pixels dans chaque bloc par ordre croissant :  $\{X_0, X_1, \dots, X_{n-1}, X_n\}$

Les informations sont récupérées à partir des blocs répondant aux conditions :

$$LM=0 \quad \text{et} \quad X_{n-1} - X_1 < T$$

On procède enfin à l'extraction des données de la charge ainsi que les LSB sauvegardés, les pixels de l'image sont restitués à leurs valeurs originales. Les équations d'extraction sont les suivantes :

$$X_n = \begin{cases} X'_n & \text{si } X'_n - X'_{n-1} = 1; b = 0 \\ X'_n - 1 & \text{si } X'_n - X'_{n-1} = 2; b = 1 \\ X'_n - 1 & \text{si } X'_n - X'_{n-1} > 2 \end{cases}$$

$$X_0 = \begin{cases} X'_0 & \text{si } X'_0 - X'_1 = -1; b = 0 \\ X'_0 + 1 & \text{si } X'_0 - X'_1 = -2; b = 1 \\ X'_0 + 1 & \text{si } X'_0 - X'_1 < -2 \end{cases}$$

$X_n$  et  $X_0$  sont les pixels restitués,  $b$  les bits de la charge et des LSB extraits.

➤ Quatrième étape : Restauration des LSB's

Les LSBs des pixels qui ont été utilisés pour la sauvegarde de l'entête sont restaurés à leurs valeurs originales.

➤ Cinquième étape : Comparatif image originale et image décodée par mesure du MSE et du PSNR

### III.4.2. La technique IPVO

La méthode IPVO est une technique qui vient améliorer la PVO, elle utilise la même méthodologie, les seules différences entre les deux techniques, résident dans l'algorithme de tri et les conditions d'intégration et d'extraction de l'information.

#### III.4.2.1. Algorithme d'intégration de données

➤ Fonction de tri :

Dans la technique IPVO, le tri doit se faire de manière particulière. Les valeurs des pixels sont triées dans un ordre croissant tout en gardant les indices de ces pixels également ordonnés dans un ordre croissant. Ceci veut dire que parmi les éléments d'un bloc, une petite valeur de pixel précède une grande valeur, mais dans le cas où les deux sont égales, la valeur avec l'indice le plus petit est mise en premier.

➤ Etape d'intégration des données :

L'intégration des données dans IPVO consiste à calculer les erreurs de prédiction  $PE_{max}$  et  $PE_{min}$ , où  $\sigma(0), \sigma(1), \dots, \sigma(n-1), \sigma(n)$  sont les indices des pixels, par :

$$PE_{max} = X_u - X_v \quad ; \quad \begin{cases} u = \min(\sigma(n), \sigma(n-1)) \\ v = \max(\sigma(n), \sigma(n-1)) \end{cases}$$

et

$$PE_{min} = X_r - X_s \quad ; \quad \begin{cases} r = \min(\sigma(0), \sigma(1)) \\ s = \max(\sigma(0), \sigma(1)) \end{cases}$$

Cette dernière manière avec laquelle sont obtenues  $PE_{max}$  et  $PE_{min}$  permet d'exploiter pour l'intégration, les deux cas où ces erreurs de prédiction sont égales à 1 ou à 0 au lieu de seulement 1 dans la méthode PVO.

Les conditions d'intégration des bits ou de décalage des pixels  $X_n$  et  $X_0$  sont données par les conditions suivantes, où  $X'_n$  et  $X'_0$  sont les résultats de l'intégration :

$$X'_n = \begin{cases} X_n + b & \text{si } (PE_{max} = 1 \parallel PE_{max} = 0) \\ X_n + 1 & \text{si } (PE_{max} > 1 \parallel PE_{max} < 0) \end{cases}$$

$$X'_0 = \begin{cases} X_0 - b & \text{si } (PE_{max} = 1 \parallel PE_{max} = 0) \\ X_0 - 1 & \text{si } (PE_{max} > 1 \parallel PE_{max} < 0) \end{cases}$$

Les bits  $b$  sont les bits de données de la charge à intégrer,  $b \in \{0, 1\}$  [18]

### III.4.2.2. Algorithme d'extraction des données

➤ Fonction de tri :

De la même manière que dans l'algorithme d'intégration, la fonction de tri est effectuée aussi en utilisant les mêmes règles.

➤ Etape d'extraction des données :

Les données sont extraites de l'image intégrée sur la base des relations :

$$X_n = \begin{cases} X_n' - 1 & \text{si } PE'max = -1, b = 1 \\ X_n' & \text{si } PE'max = 0, b = 0 \\ X_n' & \text{si } PE'max = 1, b = 0 \\ X_n' - 1 & \text{si } PE'max = 2, b = 1 \\ X_n' - 1 & \text{si } PE'max > 2 \text{ ou } PE'max < -1 \end{cases}$$

$$X_0 = \begin{cases} X_0' + 1 & \text{si } PE'min = -1, b = 1 \\ X_0' & \text{si } PE'min = 0, b = 0 \\ X_0' & \text{si } PE'min = 1, b = 0 \\ X_0' + 1 & \text{si } PE'min = 2, b = 1 \\ X_0' + 1 & \text{si } PE'min > 2 \text{ ou } PE'min < -1 \end{cases}$$

$PE'max$  et  $PE'min$ , sont les erreurs de prédictions calculées à partir de l'image intégrée,  $X_n$  et  $X_0$  sont les valeurs des pixels dans l'image décodée à partir des pixels de l'images intégrée  $X_n'$  et  $X_0'$ . La suite de bits  $b$  constitue la charge (payload) et les LSB originaux des pixels utilisés pour véhiculer les données de l'entête. [18]

### III.4.3. La technique PPVO

L'inconvénient majeur des techniques PVO et IPVO est la dépendance à la division en blocs distincts de l'image. Ceci a pour effet de diminuer considérablement la capacité d'intégration qui ne peut excéder un taux d'intégration de 0.5 bpp (pour des blocs de 2x2), du fait qu'un bloc ne peut intégrer que 2 bits au maximum. Pour pallier ce problème, la méthode PPVO est introduite. Cette technique hérite des deux premières méthodes, le fait de calculer la prédiction d'un pixel à partir des pixels voisins. Cependant elle considère chaque pixel à part afin de pouvoir éventuellement y intégrer un bit de donnée, les points voisins constituent dans ce cas un contexte pour le pixel en question.

#### III.4.3.1. Algorithme d'intégration de données

La figure III.6 représente l'organigramme de l'intégration de données utilisant la technique PPVO. [21]

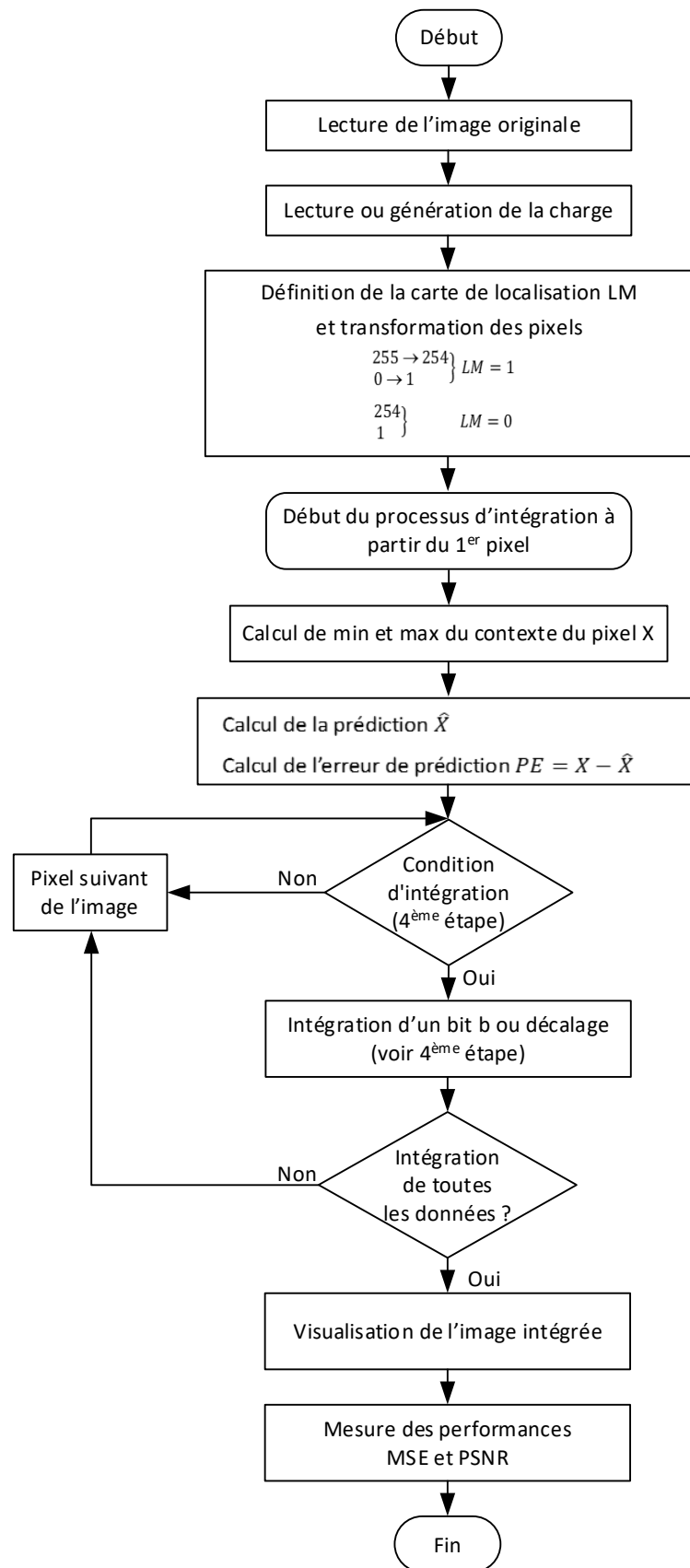


Figure III.6. Organigramme d'intégration de données par PPVO

- Première étape : Pour chaque pixel de l'image, en vue de calculer la prédiction  $\hat{X}$  du pixel  $X$ , considérer les pixels voisins qui constituent le contexte  $C1, C2, C3$  ;
- Deuxième étape : Déterminer  $\min(C)$  et  $\max(C)$ , valeurs minimales et maximales dans le contexte ;
- Troisième étape : Définition la carte de localisation
  - LM=1 si  $X=255$ ,  $X \leftarrow 254$  ( $X$  est changé en 254 pour éviter le risque de overflow)
  - LM=1 si  $X=0$ ,  $X \leftarrow 1$  ( $X$  est changé en 1 pour éviter le risque de underflow)
  - LM=0 si  $X=254$
  - LM=0 si  $X=1$

Ainsi la carte de localisation permettra de distinguer les pixels transformés en 1 ou 254 de ceux qui avaient déjà ces valeurs. Les bits constituant cette carte seront intégrés après les données de la charge.

- Quatrième étape : Insertion de la charge en suivant les étapes :
  - Calculer la prédiction  $\hat{X}$  du pixel  $X$  ainsi que l'erreur de prédiction  $PE = X - \hat{X}$  par :

Dans le cas où  $\min(C) \neq \max(C)$

$$\hat{X} = \begin{cases} \max(C) & \text{si } X \geq \max(C) \\ \min(C) & \text{si } X \leq \min(C) \end{cases}$$

Dans le cas où  $\min(X) = \max(X)$

$$\hat{X} = \begin{cases} 254 & \text{si } \max(C) = \min(C) = VC = 254 \\ VC & \text{si } \max(C) = \min(C) = VC ; (1 \leq VC \leq 253) \end{cases}$$

- Intégration des données selon les équations suivantes :

Dans le cas où  $\min(X) \neq \max(X)$

$$X' = \begin{cases} X + b & \text{si } \hat{X} = \max(C) \text{ et } PE = 0 \\ X + 1 & \text{si } \hat{X} = \max(C) \text{ et } PE > 0 \\ X - b & \text{si } \hat{X} = \min(C) \text{ et } PE = 0 \\ X - 1 & \text{si } \hat{X} = \min(C) \text{ et } PE < 0 \end{cases}$$

Dans le cas où  $\min(X) = \max(X)$

$$X' = \begin{cases} X + b & \text{si } VC = 254 \text{ et } PE = 0 \\ X - b & \text{si } VC \neq 254 \text{ et } PE = 0 \\ X - 1 & \text{si } VC \neq 254 \text{ et } PE < 0 \end{cases}$$

Les bits  $b$  sont les bits de données de la charge à intégrer,  $b \in \{0, 1\}$

- Cinquième étape : Mesure des performances MSE et PSNR

### III.4.3.2. Algorithme d'extraction des données

La figure III.7 représente l'organigramme de l'extraction de l'image originale et des données de la charge en utilisant la technique PPVO. [21]

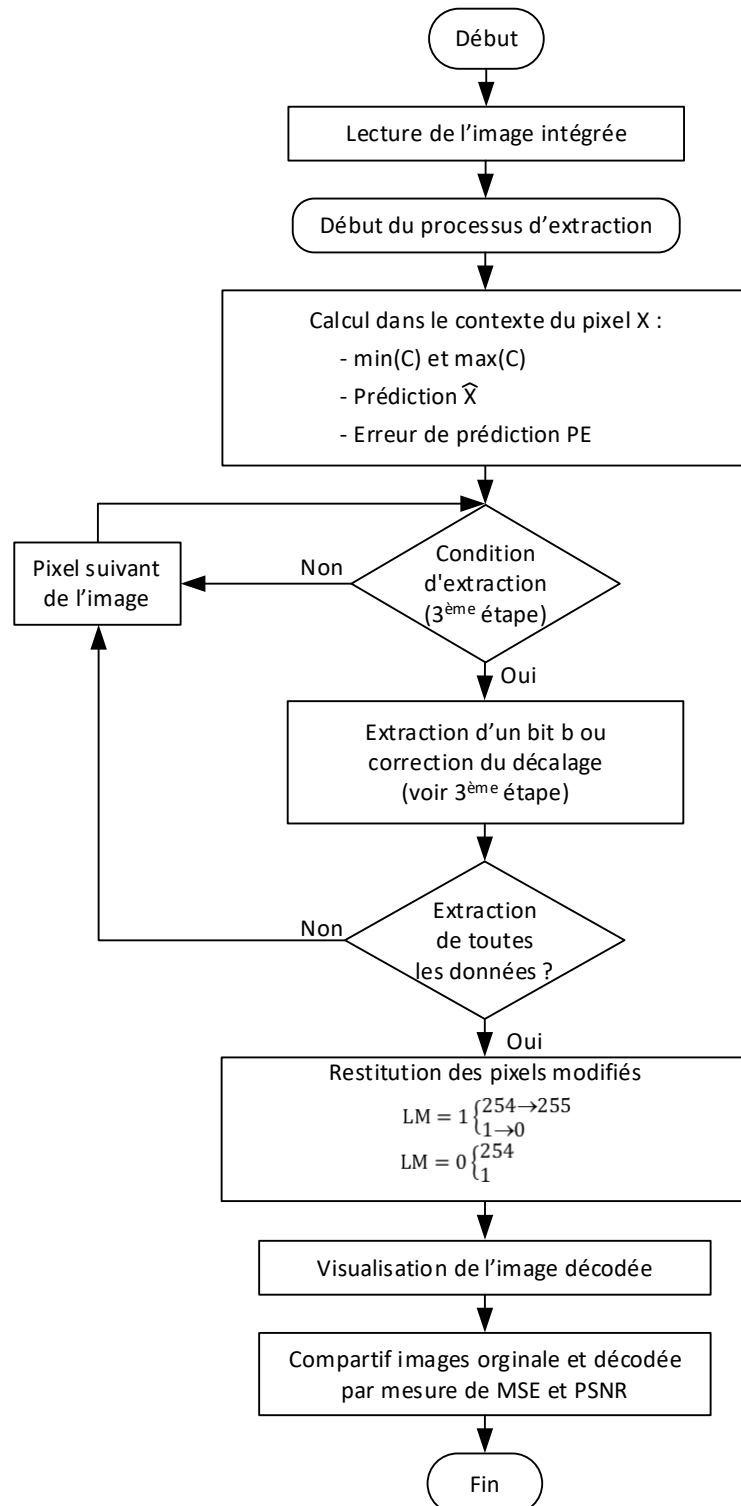


Figure III.7. Organigramme d'extraction des données par technique PPVO



- Première étape : Cibler le pixel  $X'$  de l'image intégrée pour commencer l'extraction des données, et identifier ses pixels environnants  $C1, C2, C3$  du contexte. Il est à noter qu'il est primordial de commencer par le dernier point où s'est terminé le processus d'intégration, car son contexte n'a pas été modifié. Le processus d'extraction se fera de manière séquentielle à partir de ce dernier pixel jusqu'au premier point de l'image, en restituant la valeur du pixel courant qui fait partie du contexte du pixel précédent. La position du dernier pixel est chargée dans l'image intégrée comme entête par modification de LSB.
- Deuxième étape : Déterminer  $\min(C)$  et  $\max(C)$ , valeurs minimales et maximales dans le contexte
- Troisième étape : Extraction de la charge et la carte de localisation comme suit :
  - Calculer la valeur de la prédiction  $\hat{X}$  du pixel  $X'$  de l'image intégrée et calculer l'erreur de prédiction  $PE = X - \hat{X}$ .
  - Extraction les données de la manière suivante :

Dans le cas où  $\min(C) \neq \max(C)$

$$X = \begin{cases} X' & \text{si } \hat{X} = \max(X) \text{ et } PE' = 0, b = 0 \\ X' - 1 & \text{si } \hat{X} = \max(X) \text{ et } PE' = 1, b = 1 \\ X' - 1 & \text{si } \hat{X} = \max(X) \text{ et } PE' > 1 \\ X' & \text{si } \hat{X} = \min(X) \text{ et } PE' = 0, b = 0 \\ X' + 1 & \text{si } \hat{X} = \min(X) \text{ et } PE' = -1, b = 1 \\ X' + 1 & \text{si } \hat{X} = \min(X) \text{ et } PE' < -1 \end{cases}$$

Dans le cas où  $\min(C) = \max(C)$

$$X = \begin{cases} X' & \text{si } VC = 254 \text{ et } PE' = 0, b = 0 \\ X' - 1 & \text{si } VC = 254 \text{ et } PE' = 1, b = 1 \\ X' & \text{si } VC \neq 254 \text{ et } PE' = 0, b = 0 \\ X' + 1 & \text{si } VC \neq 254 \text{ et } PE' = -1, b = 1 \\ X' + 1 & \text{si } VC \neq 254 \text{ et } PE' < -1 \end{cases}$$

La suite de bits  $b$  constitue la charge (payload) ainsi que la carte de localisation.

- Quatrième étape : Restituer à partir de la carte de localisation les pixels modifiés de 255 à 254 et de 0 à 1.
- Cinquième étape : Comparatif image originale et image décodée par mesure du MSE et du PSNR

## III.5. Conclusion

Dans ce chapitre, nous avons exposé l'interface du programme d'intégration et de décodage par les méthodes PVO, IPVO et PPVO. Nous avons mis l'accent sur les détails de programmation des algorithmes d'intégration et d'extraction de données à l'aide des trois techniques de RDH citées. Les programmes développés sous C++ Builder fonctionnent correctement et ont donné pleine satisfaction, les résultats des exécutions et des tests de performances feront l'objet du chapitre suivant.

# **Chapitre IV :**

## **Résultats et tests de performances**

## IV.1. Introduction

Vérifier le succès des études scientifiques et identifier leurs résultats réalistes nécessite des expérimentations, applications et observations, afin de prouver la justesse et l'efficacité du travail accompli.

Dans ce chapitre, on examinera la validité des trois techniques implémentées dans ce projet, à savoir les techniques PVO, IPVO et PPVO. Il s'agit également de comparer les performances de ces techniques en termes de capacité d'intégration (EC) et de rapport signal par bruit de crête (PSNR : the Peak Signal to Noise Ratio).

## IV.2. Test expérimental de l'application développée

Dans ce premier test, il s'agit de faire une expérience d'intégration d'une charge sous forme d'image ou une charge aléatoire sur une image en niveaux de gris de 512 x 512. Dans cet exemple, la technique PVO a été appliquée à l'image « Cameraman ». La figure IV.1 permet d'illustrer l'interface utilisateur du programme développé. On peut y découvrir les différents paramètres relatifs à l'opération d'intégration de données, tels que l'image de la charge ou la taille de la charge pure aléatoire, les valeurs de MSE et PSNR entre l'image originale et l'image avec données intégrées.



Figure IV.1. Exemple d'utilisation du programme d'intégration de données

Dans le processus de décodage, il est possible d'afficher des paramètres relatifs à l'opération d'extraction et des paramètres généraux tels que :

- Taux d'intégration de données en bits par pixel (bpp) ;
- Valeurs du MSE et PSNR entre l'image originale et l'image décodée ;
- Valeur de charge brute (charge pure + données additionnelles) ;
- Messages de récupération (ou d'erreur) de la charge pure et de l'image par comparaison entre les données avant l'opération d'intégration et les données après décodage ;
- Image de la charge pure (dans le cas d'utilisation d'une image comme charge) ;
- Coordonnées et valeurs de niveau de gris des pixels sur les trois images.

La Figure IV.2 représente les opérations d'intégration et d'extraction des données appliquées à l'image « Cameraman ».



Figure IV.2. L'extraction des données de Cameraman par PVO

### IV.3. Comparaison des performances des méthodes implémentées

Une analyse comparative est effectuée entre les trois méthodes implémentées (PVO, IPVO, PPVO), afin que les performances de chaque méthode puissent être déterminées et que la meilleure puisse être identifiée, grâce aux deux comparaisons de la capacité d'intégration EC et la qualité visuelle de l'image à l'aide de PSNR.

Cette comparaison est exécutée sur un groupe constitué de 5 images 512x512 de 256 niveaux de gris, largement utilisées dans le domaine des traitements d'images, ce sont les images : « Cameraman », « Jetplane », « Lena », « Baot » et « Mandril » illustrées sur la figure IV.3.



Figure IV.3. Les images du test d'intégration de données

#### IV.3.1. Comparaison de la capacité d'intégration (EC)

Les trois techniques implémentées ont été appliquées aux cinq images afin de déterminer la capacité d'intégration maximale de données  $EC_{max}$ . Les données collectées sont rassemblées sur le tableau IV.1.

Tableau. IV. 1 : Capacité d'intégration maximale  $EC_{max}$

	PVO	IPVO	PPVO
Cameraman	40037	69845	95402
Jetplane	40199	51682	62315
Lena	36165	42944	48985
Boat	23150	26452	29280
Mandrill	19681	20797	21792

La différence entre les valeurs de la capacité d'intégration entre les images, elle est le résultat de la différence de taille des zones lisses et des zones contours dans les images, de sorte que l'on constate que l'image Cameraman et Jetplane produisent une grande capacité d'intégration car elles contiennent de nombreuses zones lisses. Ce n'est pas le cas pour Mandril et Boat, elles contiennent de nombreuses zones contours et lignes de transition et sauts entre les tons de couleur, ce qui conduit à produire une faible capacité d'intégration par rapport au reste des images.

Quant à la différence de capacité d'intégration entre les techniques pour une même image, en raison du fonctionnement de chaque algorithme, les PVO et IPVO fonctionnent dans un système de blocs, c'est-à-dire qu'elles exploitent deux pixels au plus parmi les quatre pixels, contrairement à la PPVO, qui opère sur chaque pixel de l'image, ce qui a conduit à fournir une capacité d'intégration de données supérieure à PVO et IPVO, et ce pour toutes les images.

Il est clair d'après ce tableau, l'amélioration des performances en termes de capacité d'intégration apportées par les méthodes IPVO et PPVO par rapport à la méthode de base PVO. Les tests montrent également que les capacités d'intégration maximales sont obtenues par la méthode PPVO. Ces capacités d'intégration maximales sont plus grandes pour des images contenant des zones lisses importantes.

### IV.3.2. Comparaison des performances par le PSNR

Pour la comparaison des performances en termes de mesure de PSNR et de la qualité visuelle des images après intégration des données, nous avons utilisé trois capacités d'intégration de données EC communes aux cinq images (5 000 bits, 10 000 bits et 15 000 bits) en raison du fait que ces capacités peuvent être atteintes par toutes les images. Les résultats des tests de mesure de PSNR sont présentés dans les trois tableaux IV.2, IV.3, IV.4.

Tableau IV.2. Comparaison du PSNR (dB) pour EC= 5000 bits

	PVO	IPVO	PPVO
Cameraman	64.67	63.52	64.88
Jetplane	63.41	62.85	61.39
Lena	61.95	61.78	63.29
Boat	59.43	59.08	59.39
Mandril	59.51	59.12	55.98

Tableau IV.3. Comparaison du PSNR (dB) pour EC= 10000 bits

	PVO	IPVO	PPVO
Cameraman	62.56	61.62	61.86
Jetplane	60.88	60.32	58.88
Lena	59.12	58.83	59.36
Boat	56.85	56.70	56.05
Mandril	56.23	55.97	53.94

Tableau IV.3. Comparaison du PSNR (dB) pour EC= 15000 bits

	PVO	IPVO	PPVO
Cameraman	61.05	60.36	60.21
Jetplane	58.94	58.76	57.43
Lena	57.35	57.03	56.90
Boat	54.77	54.67	53.89
Mandril	54.02	53.87	52.53

En termes de qualité visuelle des images, c'est-à-dire selon les valeurs de PSNR, il ressort clairement dans les tableaux ci-dessus pour les trois différentes capacités d'intégration (5000, 10000, 15000 bits), un écart et une différence de qualité pour les trois techniques PVO, IPVO et PPVO.

Pour une quantité de données de 5000 bits, intégrée dans les images Jetplane et Boat et Mandril, le PSNR de la technique PVO est supérieur à IPVO et PPVO, tandis que dans les images de Lena et Cameraman on trouve le PSNR dans la méthode PPVO plus important que les deux autres techniques. On peut dire qu'une petite capacité d'intégration ne peut déterminer la meilleure technique en termes de qualité visuelle des images.

À mesure que plus la capacité d'intégration augmente, plus le PSNR de PVO indique la plus haute qualité entre les deux technique IPVO et PPVO, comme le montrent les tableaux



IV.3 et IV.4. Lorsque la charge est de 10 000 et 15 000 bits, PVO peut être considérée comme meilleure par rapport aux deux autres techniques IPVO et PPVO.

### **IV.3.3. Graphes de comparaison des performances**

Dans cette partie, on discutera des performances des trois méthodes d'intégration réversible de données en représentant les valeurs de PSNR par rapport à la capacité d'intégration (EC), dans des courbes graphiques illustrées dans la figure IV.4. Les graphes montrent les variations de performances PSNR des méthodes d'intégration réversible de données par rapport à la capacité d'intégration, pour chacune des images séparément, ceci pour faciliter la comparaison entre techniques par l'observation.

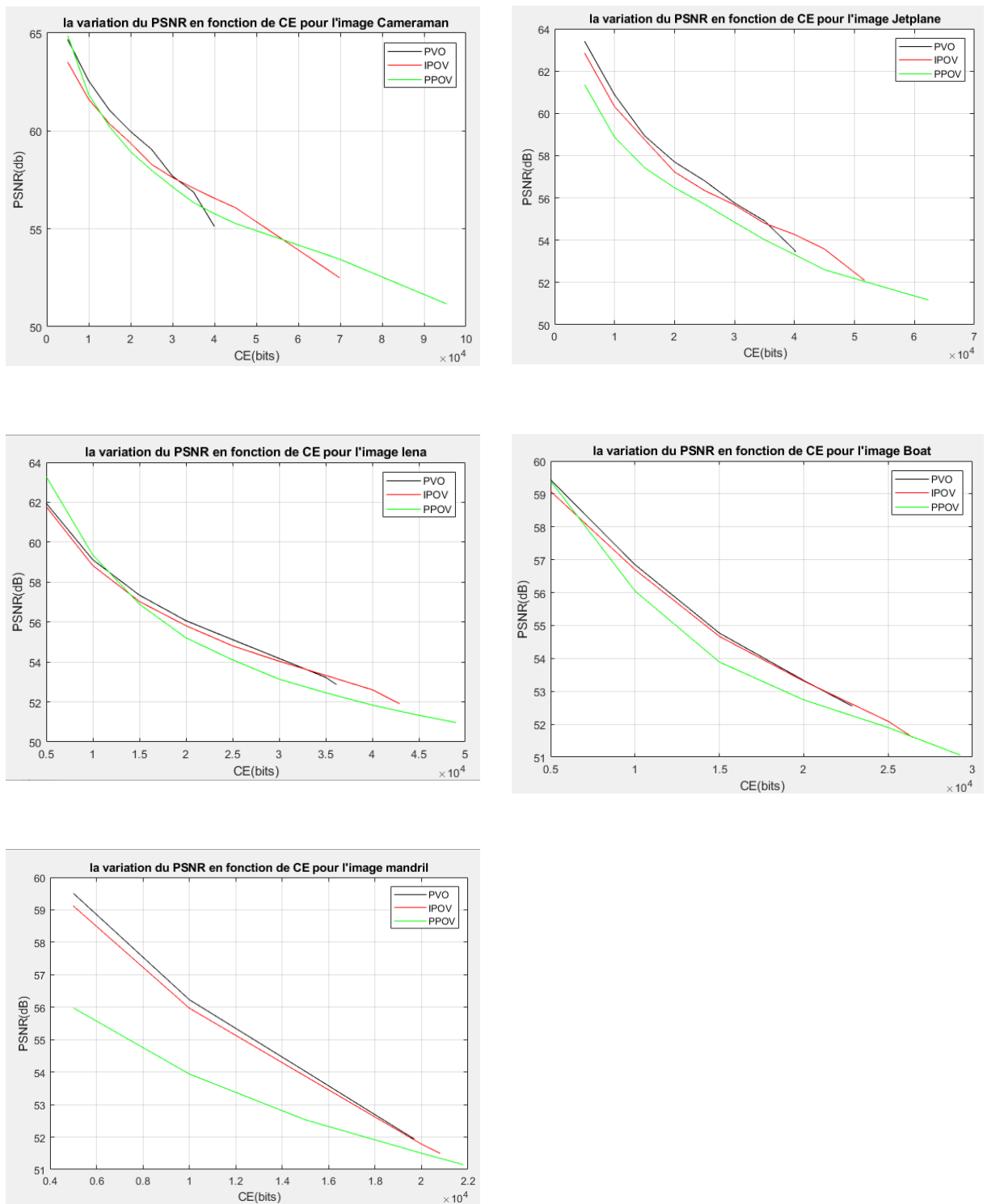


Figure IV.4. Comparaison des performances entre les méthodes PVO, IPOV et PPVO

On peut noter qu'en termes de capacité d'intégration, la méthode PPVO est la meilleure méthode qui permet d'intégrer une plus grande quantité de données par rapport aux méthodes

PVO et IPVO peu importe l'image support. Comme on l'avait mentionné précédemment, la raison est que la technique PPVO cible chaque pixel de l'image, par opposition aux deux autres méthodes qui peuvent au plus intégrer des données dans deux pixels sur quatre.

Quant au critère PSNR, les performances varient avec les images et la quantité de données à intégrer. Si on considère par exemple les images Lena et Cameraman en intégrant une petite capacité de données, la PPVO qualité apparaît meilleure (valeur élevée du PSNR) par rapport aux deux autres méthodes, Mais à mesure que la capacité augmente, les performances de la PPVO diminuent et c'est la méthode PVO devient plus intéressante. Cependant cette dernière s'arrête à une quantité limitée de donnée.

Pour ce qui est des images Jetplane, Boat et Mandril, on voit que la meilleure qualité d'image après intégration des données est obtenue par les deux méthodes PVO et IPVO pour des valeurs différentes de quantité de données, tandis que la méthode PPVO produit une image avec qualité inférieure aux deux précédents.

En conséquence, la technique PPVO est la meilleure méthode en termes de capacité élevée d'intégration des données, tandis qu'en termes de qualité de l'image apparaissant après intégration des données, la préférence des méthodes varie et est déterminée en fonction de la répartition des nouveau de gris dans l'image elle-même et la taille de la capacité de données intégrée.

## IV.4. Conclusion

Dans ce chapitre, il était question de vérifier le fonctionnement et la validité des algorithmes d'intégration et de décodage des trois techniques PVO, IPVO et PPVO dont les détails d'implémentation ont été discutés dans le chapitre précédent. Il était également question de tester les algorithmes implémentés sous C++Builder, et vérifier les performances de chaque algorithme en termes de capacité maximale d'intégration de données ( $EC_{max}$ ) et aussi en termes de qualité de l'image par mesure du PSNR. En fin, une comparaison entre les méthodes a été établie afin de définir les meilleures techniques en fonction des images et des conditions d'utilisation.

## Conclusion générale

Dans ce projet, qui porte sur l'étude et implémentation de techniques d'intégration réversible de données (RDH) basées sur la méthode PVO, nous avons étudié les avantages et applications des techniques d'intégration de données en général, et de certains des algorithmes, objets de ce travail et qui seront concernés par l'implémentation.

On a d'abord exploré l'aspect théorique, en abordant les définitions de l'intégration de données, les formes de supports et les domaines d'application, que ce soit dans le domaine des applications secrètes, ou la protection par l'utilisation de filigranes, ou le domaine des applications non secrètes. On a également découvert les types de techniques d'intégration de données non réversibles ou réversibles (RDH). Cette dernière est la plus utilisée et c'est la base de notre sujet. Les techniques DE et PEE ont été d'abord brièvement discutées, pour approfondir, par la suite, l'étude de la méthode PVO et de ses techniques dérivées, IPVO et PPVO.

Nous avons détaillé l'implémentation sous C++ Builder des algorithmes d'intégration de données des trois techniques PVO, IPVO et PPVO. Pour chacune de ces techniques, nous avons procédé à la programmation des algorithmes d'encodage et de décodage de ces trois méthodes. Les résultats des tests d'exécution ont donnée pleine satisfaction et ont permis de procéder à une étude comparative des performances de ces techniques et des conclusions ont été tirées.

Ce projet nous offre l'opportunité de perfectionner nos connaissances dans les domaines du traitement d'images, de la programmation visuelle et la programmation orientée objet. Nous avons pu pénétrer un domaine tellement important qui est le domaine de l'intégration de données.

Enfin, nous espérons que des travaux dans ce même axe de recherche verront le jour, pour étudier et implémenter d'autres méthodes et techniques d'intégration de données et proposer, pourquoi pas, de nouveaux algorithmes.

# Références Bibliographiques

[1] Husrev T. Sencar, Mahalingam Ramkumar, Ali N. Akansu, "Data Hiding Fundamentals and Applications," Elsevier Academic Press. 525 B Street, Suite 1900, San Diego, California 92101-4495, USA 84 Theobald's Road, London WC1X 8RR, UK.

[2] W. Bender, D. Gruhl, N. Morimoto, A. Lu "Techniques for data hiding," IBM SYSTEMS JOURNAL, VOL 35, NOS 3&4, 1996.

[3] Fangjun Huang, Xiaochao Qu, Hyoung Joong Kim, Jiwu Huang "Reversible Data Hiding in JPEG Images," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 26, NO. 9 SEPTEMBER 2016.

[4] Frank Y. Shih "Digital Watermarking and Steganography: Fundamentals and Techniques," Second edition, Taylor & Francis, CRC Press, 2017.

[5] Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, Ton Kalker "Digital Watermarking and Steganography," Printed in the United States of America by Elsevier, 2008.

[6] Adnan M. Alattar "Reversible Watermark Using the Difference Expansion of A Generalized Integer Transform," Tualatin, OR 97062 USA, March 3, 2003.

[7] Eugene T. Lin, Edward J. Delp, West Lafayette "A Review of Data Hiding in Digital Image," IS&T's 1999 PICS Conference.

[8] Hae Yong Kim, Sergio Vicente Denser Pamboukian, Paulo Sérgio Licciardi Messeder Barreto "Authentication Watermarkings for Binary Images".

[9] W. Bender, W. Butera, D. Gruhl, R. Hwang, F. J. Paiz, S. Pogreb "Applications for data hiding," IBM SYSTEMS JOURNAL, VOL 39, NOS 3&4, 2000.

[10] Mehmet U. Celik, Gaurav Sharma, A. Murat Tekalp, Eli Saber "REVERSIBLE DATA HIDING," IEEE ICIP 2002.

[11] Gurjinder Kaur, Samayveer Singh, Rajneesh Rani, Rajeev Kumar, "A Comprehensive Study of Reversible Data Hiding (RDH) Schemes Based on Pixel Value Ordering (PVO)," CIMNE, Barcelona, Spain 2020, Archives of Computational Methods in Engineering (2021) 28:3517–3568

- [12] Ziming Zhang, Wenjia Wang, Zihan Zhao, Erfu Wang, "PVO-based reversible data hiding using bit plane segmentation and pixel expansion," *Journal of Information Security and Applications* 79 (2023) 103649, Elsevier Ltd, 4 November 2023.
- [13] Weiqing Wang, Junyong Ye, Tongqing Wang, Weifu Wang, "Reversible data hiding scheme based on significant-bit-difference expansion," *IET Image Process.*, 2017, Vol. 11 Iss. 11, pp. 1002-1014, the Institution of Engineering and Technology 2017.
- [14] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans Circuits Syst Video Techn.*, vol. 13, no. 8, pp. 890–896, 2003.
- [15] Dwi S. Angreni, Tohari Ahmad "Enhancing DE-based Data Hiding Method by Controlling the Expansion" Department of informatics Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur 60111 Indonesia.
- [16] M. Weinberger, G. Seroussi, and G. Sapiro, "LOCOI: A Low Complexity, Context-Based, Lossless Image Compression Algorithm:" in *Proc. of the IEEE Data Compression 1996*, pp. 140-149.
- [17] M. Kumar and S. Agrawal "Reversible data hiding based on prediction error Expansion using adjacent pixels " *Security and Communication Networks*, 2016 • Wiley Online Library
- [18] Fei Peng, Xiaolong Li, Bin Yang, "Improved PVO-based reversible data hiding," *Digital Signal Processing* 25 (2014) 255–265, Elsevier 2013.
- [19] Bo Ou, Xiaolong Li, Yao Zhao, Rongrong Ni, "Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion," *Signal Processing: Image Communication* 29 (2014) 760–772, Elsevier 2014.
- [20] Xiaolong Li, Jian Li, Bin L, Bin Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," *Signal Processing* 93 (2013) 198–205, Elsevier 2012.
- [21] Xiaochao Qu, Hyoungh Joong Kim, "Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding," / *Signal Processing* 111 (2015) 249–260, Elsevier 2015...
- [22] <https://www.embarcadero.com/>
- [23] [\\_help. C++Builder](#)