INTERPOLATION BASED ADAPTIVE REVERSIBLE DATA HIDING FOR DIGITAL IMAGE APPLICATIONS

MD. ABDUL WAHED

Student No - 1014160009 (P)

A THESIS SUBMITTED IN FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL, ELECTRONIC AND COMMUNICATION ENGINEERING

DEPARTMENT OF
ELECTRICAL, ELECTRONIC AND COMMUNICATION ENGINEERING

MILITARY INSTITUTE OF SCIENCE AND TECHNOLOGY
MIRPUR CANTONMENT, DHAKA 1216

NOVEMBER 2018

APPROVAL CERTIFICATE

The thesis titled "**Interpolation based Adaptive Reversible Data Hiding for Digital Image Applications**" submitted by Md. Abdul Wahed, Roll No: 1014160009(P), Session: 2014-2015 has been accepted as satisfactory in partial fulfilment of the requirements for the degree of Master of Science in Electrical, Electronic and Communication Engineering on 15 November 2018.

BOARD OF EXAMINERS

1. _____ Chairman
   Major Hussain Md. Abu Nyeem, PhD, EME (Supervisor)
   Instructor Class B
   Department of EECE, MIST, Dhaka - 1216

2. _____ Member
   Colonel A K M Nazrul Islam, PhD (Ex-officio)
   Head
   Department of EECE, MIST, Dhaka - 1216

3. _____ Member
   Air Commodore Md Hossam-E-Haider, PhD (Internal)
   Professor
   Department of EECE, MIST, Dhaka - 1216

4. _____ Member
   Dr. Satya Prasad Majumder (External)
   Professor
   Department of EEE, BUET, Dhaka - 1205

CANDIDATE'S DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Signature: _____

(Md. Abdul Wahed)

Date: November 2018

DEDICATION

To my parents and family

ACKNOWLEDGEMENTS

The accomplishment of my M.Sc. research and write-up of this thesis are the most memorable events in my life so far, which might not be possible without the support of my supervisor, colleagues, friends, and family. At the end of this MSc research, it is a pleasure to thank everybody, who has helped me all the way.

First and foremost, I would like to thank my supervisor, Major Hussain Md. Abu Nyeem, PhD, EME, who has been a perfect mentor for me. I am grateful to him for giving me directions, confidence and valuable advice, and for everything that I have learned from him. He has had a tremendous impact on the way I perceive research and life in general. Major Nyeem has also been very helpful in many ways to me and for my research. His feedback from different critical angles has always been encouraging and useful. He also has taught me many practical aspects of research from the elementary level. Once again I thank my supervisor, with whom I feel that I genuinely have been privileged to work.

I profoundly thank Professor Satya Prasad Majumder; Air Commodore Md Hossam-E-Haider, PhD, BAF and Colonel A K M Nazrul Islam, PhD for their valuable and constructive feedback, assistance and being on my final defense panel. I thank Major Md Tawfiq Amin, PhD, EME for kind coordination and cooperation as the post-graduate course coordinator. Also thanks to all the reviewers, for their appreciation and feedback on my thesis and publications.

I am grateful to my colleagues and friends for making my working at MIST and BMA, more than comfortable and enjoyable. I express my gratitude for their suggestions and help to find a solution, technical or otherwise. I would be pleased to extend my sincere thanks to all of my course teachers and staffs of EECE department, MIST for their kind support. Special thanks to Captain Mahbub, Captain Shafiul and Mr. Hafizur Rahman for their cordial assistance and official support to complete the administrative procedures.

I thank each member of my family, for being understanding and supportive. I am most thankful to my parents, for their invaluable encouragement. My mother, in spite of her sickness from the last two years, who always encourages me all the ways, I can not express my gratitude to her in words. Similarly, I could not find words that express my gratefulness to my wife Farhat Ferdous Bonny, for her supportive and understanding attitude. Many conversations, laughs and sweet memories with her helped take the pressure off during my stressful times. Furthermore, I would also like

to thank Mrs. Luisa Fozila Chowdhury, the spouse of my supervisor, for her endless patience during my working days and nights with my supervisor. She was always loving and caring to me that never made me feel away from home.

Although many of you may not understand, nor ever want to understand, much of what is written beyond this page, you should at least know that it could not have been done without you.

MD. ABDUL WAHED

Military Institute of Science and Technology

Dhaka, Bangladesh

November 2018

ABSTRACT

Reversible Data Hiding (RDH) has emerged to be a significant multi-disciplinary research area. Different types of RDH schemes have already been developed and demonstrated their potentials for applications like media-content authentication, copy-right protection, integrity establishment and annotation. Recently, interpolation based RDH (IRDH) schemes have shown great promises for better rate-distortion performance with a set of unique advantages over their counterparts. Despite great promises, existing IRDH schemes, however, have no consideration of 'effective' capacity management for varying size of the payload. As a result, disregarding the varying embedding-capacity requirement, existing IRDH schemes fail to maintain the minimum possible distortion in the embedded image. An IRDH scheme with adaptive embedding is therefore vital for better rate-distortion performance.

In this thesis, a new IRDH scheme is developed and presented based on a simplified parabolic interpolation (SPI) and adaptive embedding. Additionally, an adaptive embedding approach is introduced with two new techniques, where a capacity control parameter is formulated and utilized to attain the varying capacity requirement by determining a minimum set of embeddable bits in a pixel for the best possible embedded image quality. The first technique introduces a flag bit for distinguishing the original payload bits from their complement which are embedded in the interpolated pixels. In the second embedding technique, the correlation between an embeddable pixel and estimated versions of an embedded pixel is utilized to avoid the use of any flag bit as such an extra payload bit per pixel can be embedded therein. Thereby, the embedding capacity significantly increases without compromising the image quality for similar values of the capacity control parameter.

Computational modeling of the new IRDH scheme is presented and its performance is evaluated with a set of popular test-images. While compared with the prominent IRDH schemes, the proposed scheme both with or without flag demonstrated its efficiency for significantly better embedding rate-distortion performance. Moreover, being up-sampled, the embedded image would have a higher spatial resolution. It also does not require any location map, and thus the total capacity can be effectively used for data embedding. Nevertheless, it keeps the original pixels untouched and thus, would be useful in military and medical image applications that restrict minimum possible changes in the cover images.

KEYWORDS

Adaptive embedding
Capacity control parameter
Computational efficiency
Content authentication
Copyright
Embedding
Embedding capacity
Embedding rate
Image recovery
Information management
Interpolation
Interpolation based reversible data hiding
Medical information
Patients' medical records
Payload extraction
Payload
Peak signal to noise ratio
Rate-distortion performance
Reversible data hiding
Simplified parabolic interpolation
Structural similarity
Visual image quality

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

BI	Bilinear Interpolation
DE	Difference Expansion
DLE	Direct LSB Embedding
ENMI	Enhanced Neighbor Mean Interpolation
INP	Interpolation by Neighbor Pixels
IRDH	Interpolation based Reversible Data Hiding
LSB	Least Significant Bit
MSB	Most Significant Bit
MSE	Mean Squire Error
NMI	Neighbor Mean Interpolation
NNI	Nearest Neighbor Interpolation
PEE	Prediction Error Expansion
PI	Parabolic Interpolation
PSNR	Peak Signal to Noise Ratio
RCM	Reversible Contrast Matching
RDH	Reversible Data Hiding
SPI	Simplified Parabolic Interpolation
SSIM	Structural Similarity

LIST OF SYMBOLS

I	an input image
C	an interpolated image
I_{up}	an initialized up-sampled image, *i.e.*, $I_{up} - I = \{0\}$
I'_{up}	an up-sampled image with interpolated pixels
I_{em}	an embedded image
y	an image pixel
y'	an embedded pixel
L	bit-depth of an image, $L \in \mathbb{N}$
n	bit-length of any pixel
T	capacity control parameter (*i.e.*, number of unchanged MSBs in $y \in C - I$)
a, b, c	coefficients of a parabolic equation
g	direct interpolated value in parabolic interpolation (PI)
γ	embedding capacity requirement (*i.e.*, size of payload in bits, $\gamma = \|\|data\|\|$)
n_{em}	number of embeddable bits of an interpolated pixel
M	number of pixels in a row: $M \in \mathbb{N}$
N	number of pixels in a column: $N \in \mathbb{N}$
d	pixel difference
(i, j)	pixel position
$data$	payload , $\{0, 1\}^{\gamma}$
E_c	total embedding capacity in bit
F	up-sampling factor, $F \in \mathbb{N}$
y_i, $y_{i,j}$	y of i^{th} or $(i, j)^{th}$ position in 1D or 2D array, respectively, which also applies to y'
I_{Hblock}, I'_{Hblock}	1×5 size image block of horizontal pixels: $I_{Hblock} \subset I_{up}$ and $I'_{Hblock} \subset I'_{up}$
I_{Dblock}, I'_{Dblock}	1×5 size image block of diagonal pixels: $I_{Dblock} \subset I_{up}$ and $I'_{Dblock} \subset I'_{up}$
I_{Vblock}, I'_{Vblock}	1×5 size image block of vertical pixels: $I_{Vblock} \subset I_{up}$ and $I'_{Vblock} \subset I'_{up}$

CHAPTER 1

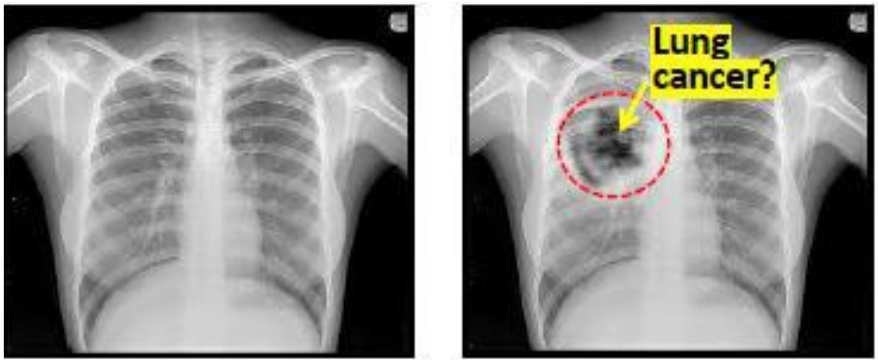
INTRODUCTION

1.1 Introduction

All those we see may not always be true. For example, consider the images shown in Fig. 1.1. These images although look very real, actually they are not, since a dog can not have a head of a bird nor a bird can have a head of a dog. So, we know the reality and can easily say that these images are not real, they are modified. Such manipulation in digital images has become much easier using commonly available applications and devices like our mobile phones. The modified images although look very real, they can be undetectable by mere visual inspection.

Particularly, when the reality is unknown, we cannot say for sure whether an image has been modified or not. For instance, in the case of medical images, the affected regions can be very easy to modify, but it might be quite impossible to detect and locate the modification with any conventional tools. Such manipulation may be done only for fun or with some illicit intention such as for false insurance claim, false evidence or other fraudulent activities [1–6]. Clinical use of the manipulated images can

Figure 1.1: Example of manipulated images (Courtesy: `www.worth1000.com`)



Original image Tampered image

Figure 1.2: Example of medical image manipulation

further be life-threatening in case of medical images. As a result, the reinforcement of the authenticity of digital images has been an open challenge today that requires some special security tools. Reversible Data Hiding (RDH) is one way to address this growing security concern, which has been widely investigated in recent years.

1.1.1 Reversible data hiding (RDH)

Reversible Data Hiding (RDH) has emerged to be a significant multi-disciplinary research area that captures different applications of signal processing, big-data and biotechnological applications, multimedia security, and information management [7–11]. An RDH scheme generally embeds a set of data (or more specifically *payload* that includes message or watermark with any side information) into a cover image (or other media like *audio* and *video*) so that the embedded data can be exactly extracted followed by a complete recovery of the cover image. Thus, it has three main processes: *generation*, *embedding* and *extraction* [12, 13]. In the *generation*, required secret-data are generated for an intended application such as content authentication, copyright protection, annotation, *etc*. The *embedding* process, on the other hand, deals with how and where the data are to be embedded in the cover media. The *extraction* process later extracts the embedded data and completely recovers the original media.

1.1.2 RDH requirement

Many RDH schemes have been developed for different applications like media-content authentication [14, 15], copyright protection [7, 16, 17], integrity establishment [18, 19] and annotation [20–22]. Particularly, for digital image applications that the research presented in this thesis mainly focuses on, RDH schemes are evolving to attain better *embedding rate-distortion* performance. Two main requirements are thus generally attained: (*i*) better embedded image quality and (*ii*) higher embedding capacity. Data embedding introduces an inevitable distortion to the cover image. Minimizing this distortion also reduces the embedding capacity. As a result, achieving high embedding capacity (or, rate) with low distortion has remained challenging, addressing which underpins many data hiding schemes.

1.2 Literature Review

Many reversible data hiding schemes have been developed for higher embedding capacity with lower embedding distortion (or better image quality). For example, Direct Least Significant Bits Embedding (DLE) schemes [23–25], Difference Expansion (DE) schemes [26–29], Histogram Shifting (HS) schemes [30–34]; Reversible Contrast Matching (RCM) schemes [35–38], Prediction Error Expansion (PEE) schemes [39–44] are a few to name.

Recently, interpolation based RDH (IRDH) schemes have created a new paradigm in data hiding research [25, 45–51]. Fig. 1.3 shows a general framework of IRDH schemes having two main processes: (i) interpolation for up-sampling (ii) embedding for data hiding. A cover image is up-sampled using interpolation. (We now and onward call the up-sampled image an interpolated image.) Then the payload is embedded into the interpolated pixels of the up-sampled image with a suitably designed reversible

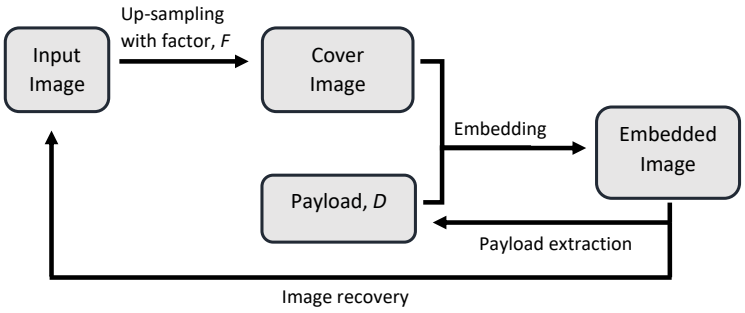


Figure 1.3: General framework for IRDH schemes.

embedding technique. Thus, the development of IRDH schemes mainly aims at the improvement of two underlying processes: (*i*) computation of interpolated pixels and (*ii*) embedding of payload.

For computing interpolated pixels, different interpolation techniques are employed. Thereby, the cover image is up-(or down-) sampled. Commonly used interpolation techniques include neighbor mean interpolation [45], neighbor pixels interpolation [46], combination of interpolation and HS [47], combination of maximized difference between neighbor pixels and an extended interpolation technique [52], and parabolic interpolation in all possible directions of up-sampled pixels in an image [49].

For embedding data, on the other hand, interpolated pixels of an input image are generally used, keeping the original pixels untouched that are desirable in military and medical image applications. Embedding techniques in the IRDH schemes can broadly be classified into three groups; namely (i) for computing prediction errors (PEE-based schemes [40, 41, 53]), (ii) for computing embeddable bits (DLE-based schemes [25, 45, 46]), and (iii) combination of both [48]. Due to some unique advantages of DLE-based IRDH schemes over its counterparts (see Sec. 2.3), the research presented in this thesis sets its goal to the development of a DLE-based IRDH scheme. Particularly, in some reversible data hiding applications, for example, annotation of patients electronic medical records, the payload size is usually much higher. In those applications, attaining significantly higher embedding capacity requirement is more than

challenging. So, the DLE-based IRDH schemes are recently investigated for higher embedding capacity that embed in the interpolated image (i.e., up-sampled cover image) for effective management of the large database of medical images and associated patient information.

However, despite their great promise to the applications requiring high embedding capacity, the existing IRDH schemes do not account for any varying capacity requirement (see Sec. 2.4). Embedding of varying size payload (*i.e.*, attaining varying capacity requirement) is a significant data hiding problem for digital image applications, which has not been addressed yet. The main reason is that payload size usually varies in many data hiding applications including those mentioned above. Thus for embedding of varying size payloads, the distortion in the embedded images remains uncontrolled. It is generally assumed that embedding continues for the last bit of payload resulting in non-uniform distribution of distortion in the embedded image. In other words, for embedding of varying size payloads, existing schemes may result in the same level of distortion without maintaining the best possible rate-distortion performance for different size of payloads within the maximum capacity limit. One way to tackle this embedding problem is to use a suitably designed adaptive data embedding technique.

Thus, for varying embedding capacity requirement, if a uniform embedding distortion is always ensured, embedded image quality can be further improved. So the problem is "how can we achieve that?". We need to make the embedding adaptive, which the research presented in this thesis aims to develop, employ and validate in designing a new IRDH scheme. Therefore, it is necessary and makes sense to investigate the systematic development and evaluation of RDH schemes in general for digital image applications, addressing the identified research gaps.

1.3 Research Objectives

The research presented in this thesis sets its primary goal to develop an adaptive embedding technique and utilize it in modelling a new IRDH scheme as mentioned in the previous section.

To carry out the project, the specific objectives of this work are outlined as follows.

a. **To analyze existing potential interpolation techniques used in state-of-the-art RDH Schemes and thus, to determine or develop a suitable interpolation technique that offers better image quality with lower computational complexity.**

b. **To investigate the effect of different LSBs of the interpolated pixels on the embedded image quality and thereby, to define a generalized capacity control parameter.**

c. **To develop the algorithms of the data embedding and extraction processes for complete formulation of the proposed RDH scheme.**

d. **To develop an experimental setup for performance evaluation and benchmarking of the proposed scheme to demonstrate the expected embedding rate-distortion efficiency for varying size of payloads.**

The expected outcome of the proposed research is an adaptive RDH scheme that could effectively manage the rate-distortion performance for varying embedding capacity requirements of digital image applications.

1.4 Research Questions

Development of an IRDH scheme may have several considerations, for example, design requirements and performance requirements. Design requirements include how the interpolation and embedding techniques would be designed, where the data would

be embedded, how it affects the original pixels *etc*. Performance requirements include computational complexity of the processes and the visual quality of the interpolated and embedded images for varying size payloads, the embedding capacity (*i.e*. the rate-distortion performance) *etc*. To be specific to the research problem, the following research question can be focused.

Can an adaptive IRDH scheme be developed to effectively manage the varying capacity requirement for better image quality?

The question leads us to address two main challenges: choosing a suitable interpolation technique and developing an adaptive embedding. To address these challenges, few sub-questions also arise as follows.

(i) ***Which interpolation technique is better for up-sampling an input image that can offer the best possible interpolated image quality?***

(ii) ***How can we achieve the varying capacity requirement for embedding in IRDH scheme****?*

(iii) ***How can a better image quality be assured with the adaptive embedding?***

The first sub-question entails an investigation among the existing interpolation techniques aiming at identifying the one for the best possible interpolated image quality. All the interpolation techniques, moreover, may not be equally computationally efficient. Selecting an appropriate interpolation technique is required prior to the employment of it in IRDH schemes for data embedding.

Additionally, it is not only the interpolation but also an effective embedding technique which affects the embedding-rate distortion performance of an IRDH scheme. As mentioned in Sec. 1.2, the thesis focuses on the development of IRDH schemes with DLE. However, the existing IRDH schemes with DLE do not account for embedding

different size payloads for attaining varying capacity requirement without compromising the maximum possible embedding rate and image quality. Consideration varying capacity requirement is important to maintain the uniform embedding distortion in an embedded image. This brings about the requirement of development of a new adaptive embedding technique for IRDH scheme.

Moreover, to maintain the better image quality with adaptive embedding is also very important, thus arises the third sub-question. A generalized capacity control parameter can be formulated which would adaptively be selected utilizing the image size, up-sampling factor and the payload size (*i.e.*, capacity requirement) for embedding a minimum number of LSBs to ensure minimum possible distortion.

1.5 Research Scope

(*i*) Among enormous multidisciplinary applications of RDH schemes, the research presented in this thesis limits its focus on the digital image applications only.

(*ii*) Any data, images, processes, algorithms, schemes or methods considered in this thesis either are digital themselves, or deal with digital inputs and outputs. However, for readability, the word *digital* is often omitted in writing. For example, *digital watermarking*, *digital medical image*, *etc*. are merely written as *watermarking*, *medical image*, *etc*., respectively without change of their meaning and context.

(*iii*) Many RDH schemes have been developed for different digital image applications. Recently, IRDH schemes have been widely investigated to attain better rate-distortion performance. So, the research conducts a thorough review of the existing IRDH schemes.

(*iv*) Due to some unique advantages of IRDH schemes with DLE over their coun-

terparts, the thesis again limits its scope on the development of the DLE-based IRDH schemes. In spite of their great promises, the existing DLE-based schemes have no specific consideration of embedding varying size payloads. Thus the research addresses a significant data hiding problem of varying capacity requirement in Sec. 1.1. This research is focused on developing an adaptive IRDH scheme to attain the varying capacity requirement without compromising in the embedding distortion in many practical digital image applications like annotation of patients electronic medical records.

1.6 Research Outcomes

In addressing the research objectives and research questions, this thesis presents a number of original contributions and achievements in the field of reversible data hiding. The research contributions and findings that will be discussed in this thesis have been presented in several reputable conferences, and published and submitted to leading journals (see the list of publications in Page 78). The primary outcome of the research presented in this thesis includes a new IRDH scheme with two adaptive embedding techniques. The key contributions are summarized below.

(*i*) Developed and presented a new adaptive IRDH scheme with the following features:

- A new simplified parabolic interpolation technique is developed for better computationally efficiency.
- A new adaptive embedding is developed utilizing a general capacity condition and a generalized capacity control parameter for adaptive embedding.

- The improved visual quality of the embedded image is ensured utilizing the complement of the embeddable payload. In embedding, LSBs are used for embedding payload either in their original or complement form for minimizing embedding distortion. (See Chapter 3, Sec. 3.4.1)
- A flag bit is used to track whether the embedded data is original or its complement required for extracting the payload.
- High embedding capacity with better embedded image quality is obtained, which outperforms the existed DLE based IRDH schemes. However, the embedding capacity could further be significantly increased if the use of flag bit is avoided.

(*ii*) Further improvement in the embedding is attained utilizing the correlation between the embeddable pixels and embeddable payload bits. The said correlations reinvestigated to improve the embedding capacity without compromising the embedded image quality. Consequently, a simplified IRDH scheme with a new adaptive embedding without flag is developed and evaluated, where:

- The capacity condition and the capacity control parameter are more formally defined with the generalized algorithmic detail.
- The use of flag bit is avoided by utilizing the correlation between the embeddable pixel and estimated versions of embedded pixel.
- The rate-distortion performance of the proposed scheme is evaluated and validated for the improved embedding capacity with better visual quality. (See Chapter 3, Sec. 3.4.2)

It is noted here that, from now on, throughout the whole write-up, the new IRDH scheme developed in this thesis is addressed as "the proposed" scheme. Similarly, the new adaptive embedding techniques are also addressed as the proposed embedding techniques.

1.7 Research Significance

This research advances knowledge in the area of reversible data hiding and its application to digital images. The potential significance of the research is noted below.

(*i*) We have developed and presented an adaptive IRDH scheme, which can deterministically embed different size payload in an image with a suitable value of the capacity control parameter. In other words, the proposed scheme adaptively selects the value of the parameter for attaining varying embedding capacity requirement with a minimum possible distortion in the embedded image.

(*ii*) Embedding process is modeled to utilize the logical correlation between the embeddable pixel and estimated versions of the embedded pixel to increase the embedding capacity.

(*iii*) Extraction is modeled to blindly extract the embedded payload and to completely recover the original image, where these two processes were kept mutually independent to ensure better user control in an application scenario. This also means that the cover image can be instantly recovered with or without extracting the embedded payload.

(*iv*) Experimental results demonstrated the effectiveness of the proposed scheme for better visual quality with higher embedding capacity compared to other IRDH schemes.

(*v*) The original pixels of a cover image are kept untouched and thus, the proposed scheme would be useful in military and medical image applications that restrict minimum possible changes in a cover image.

(*vi*) Being up-sampled, the embedded image would have a higher spatial resolution. It also does not require any location map, and thus the total capacity can be effectively used for data embedding.

1.8 Thesis Organization

The remainder of this thesis is organized as follows.

Chapter 2 captures the background of the proposed research of this thesis. With an overview of existing prominent IRDH schemes, their associated techniques are reviewed and the key potentials and limitations are studied. The scopes of improvement in IRDH schemes are investigated. Thereby, a further study towards developing an adaptive IRDH scheme is addressed.

Chapter 3 presents the gradual development of a new adaptive IRDH scheme with specific algorithmic details of a computationally efficient simplified interpolation, the adaptive embedding and extraction technique. The formal definition of the embedding condition and capacity control parameter is also developed. The work presented in this chapter has been supported by meaningful examples, necessary explanations, and comparative studies.

Chapter 4 presents the evaluation of the proposed scheme for its embedding rate - distortion performance. The performance of the proposed scheme is validated in this chapter with high capacity and better image quality compared to prominent IRDH schemes.

Chapter 5 presents the conclusion of the thesis with a summary of the original contributions and future work.

CHAPTER 2

RELATED IRDH SCHEMES

2.1 Introduction

This chapter captures the background of the research contribution presented in this thesis. A thorough review of the existing interpolation techniques is given in Sec. 2.2 and their uses for data embedding in the existing prominent IRDH schemes in Sec. 2.3. The key potentials and limitations of the existing IRDH schemes are studied and the scopes of improvement in IRDH schemes towards the development of a new adaptive IRDH scheme are investigated at the end of the chapter (See Sec. 2.4).

As discussed in Chapter 1, an IRDH scheme has two main processes: (i) interpolation for up-sampling (ii) embedding for data hiding. It is to mention that interpolation techniques are chosen to obtain an up-sampled image containing both the original and interpolated pixels. Since the image quality of the up-sampled image eventually contributes to maintaining better embedded image quality, lower distortion in the up-sampled image is always desirable. We note that for evaluating distortion between an original image and its interpolated version, the original image is first down-sampled followed by the up-sampling using the interpolation technique in question. Unlike this, in an IRDH scheme, an input image is directly up-sampled (using interpolation) for embedding, where interpolated pixels are used in different ways (for example, PEE and DLE as mentioned in Sec. 1.1), to achieve better rate-distortion performance, which is discussed in the following sections.

2.2 Interpolation Techniques

As we mentioned in Chapter 1, different interpolation techniques have been employed in IRDH schemes for up-sampling the cover image before embedding. For example, neighbor mean interpolation (NMI) [45], interpolation by neighboring pixels (INP) [46], enhanced neighbor mean interpolation (ENMI) [48] and parabolic interpolation (PI) [49] are a few that showed great promises for IRDH schemes. Additionally, bilinear interpolation (BI) and nearest neighbor interpolation (NNI) are two basic interpolation techniques that have not been employed in IRDH schemes, but they are used to compare the performance of interpolation in IRDH schemes (for example, see [49, 50]). So, in this section, the technical details of the prominent interpolation techniques are discussed as follows.

2.2.1 Bilinear Interpolation (BI) and Nearest Neighbor Interpolation (NNI)

BI technique estimates an interpolated pixel considering its four nearest original pixels. An weighted average of the four pixels is computed as the interpolated pixel value [50]. A 2×2 nearest neighborhood of an interpolated pixel are considered and their weighted average is computed for estimating the interpolated pixel. For example, see Fig. 2.1(a), where the interpolated pixels $C(1,1)$ and $C(1,2)$ are calculated by weighting its neighbors $C(0,0)$, $C(2,0)$, $C(0,3)$ and $C(2,3)$, which are the original pixels of the cover image, I. That is, $C(0,0) = I(0,0), C(0,3) = I(0,1), C(2,0) = I(1,0)$ and $C(2,3) = I(1,1)$. Additionally, $C(1,1) = \frac{2}{3}[\frac{C(0,0)+C(2,0))}{2}] + \frac{1}{3}[\frac{C(0,3)+C(2,3)}{2}]$ and $C(1,2) = \frac{1}{3}[\frac{C(0,0)+C(2,0))}{2}] + \frac{2}{3}[\frac{C(0,3)+C(2,3)}{2}]$ are the interpolated pixels.

On the other hand, NNI determines the nearest neighboring pixel as an interpolated pixel. For example, refer to Fig. 2.1(b), the pixels of an interpolated image, C are obtained from their respective nearest original pixels of the cover image I. Thus,

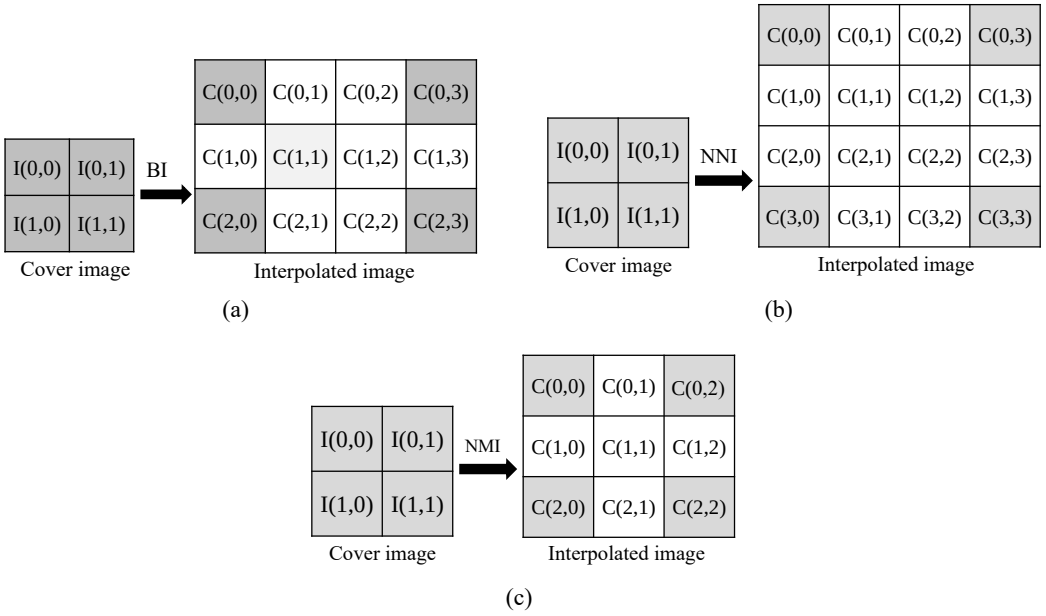


Figure 2.1: Example of computing interpolated pixels using: (a) BI, (b) NNI, and (c) NMI.

$C(0,0) = C(0,1) = C(1,0) = C(1,1) = I(0,0)$; $C(0,3) = C(0,2) = C(1,2) = C(1,3) = I(0,1)$; $C(3,0) = C(2,0) = C(2,1) = C(3,1) = I(1,0)$; $C(3,3) = C(2,2) = C(2,3) = C(3,2) = I(1,1)$. These two techniques are simple, but they produce a low quality interpolated image with mosaic and saw tooth artifacts [54]. They are therefore not directly used to compute the up-sampled image, rather they are usually used for comparing up-sampled image quality in different IRDH schemes.

2.2.2 Neighbor Mean Interpolation (NMI)

The NMI technique [45] computes an interpolated pixel as the average of the nearest original pixels. For horizontal or vertical neighborhood, two nearest pixels are considered and for the diagonal neighborhood, three nearest pixels are considered. Thereby, a 2×2 block of an input image is up-sampled to a size of 3×3 containing five interpolated pixels.

A given input image, I of a size $\frac{M}{2} \times \frac{N}{2}$is up-sampled with NMI using Eq. 2.1 and 2.2 to obtain the interpolated image, C of size $M-1 \times N-1$. For example, Fig. 2.1(c)

demonstrates how a 2×2 block of I is up-sampled using NMI to a 3×3 block of C, where $C(0,0) = I(0,0),\ C(0,2) = I(0,1),\ C(2,0) = I(1,0)$, and $C(2,2) = I(1,1)$ are the original pixels and the others are the interpolated pixels.

$$C(i,j) = I(\tfrac{i}{2}, \tfrac{j}{2}) \quad \text{if } (i,j) \bmod 2 = 0 \tag{2.1}$$

$$C(i,j) = \begin{cases} \frac{C(i,j-1)+C(i,j+1)}{2} & \text{if } i \bmod 2 = 0, j \bmod 2 = 1 \\ \frac{C(i-1,j)+C(i+1,j)}{2} & \text{if } i \bmod 2 = 1, j \bmod 2 = 0 \\ \frac{C(i-1,j-1)+C(i-1,j)+C(i,j-1)}{3} & \text{otherwise} \end{cases} \tag{2.2}$$

While this offers reasonably good up-sampled image quality, reconsidering the neighborhood and interpolated pixels for computing the final interpolated pixels, a better quality up-sampled image can be obtained as reported in INP [46] and ENMI [48].

2.2.3 Interpolation by Neighboring Pixels (INP)

Lee & Huang [46] proposed an interpolation by neighbor pixels (INP) is an improved version of the NMI technique that utilizes the full advantages of the neighborhood pixels. Unlike NMI, this technique considers all the original neighborhood pixels. For example, for computing either a horizontal or vertical interpolated pixels, two nearest original pixels, one in a vertical neighborhood and another in a horizontal neighborhood are used. For the horizontal interpolated pixel, weight for the horizontal neighborhood pixel is set to 0.75 and that for the other is set to 0.25. The weights are altered accordingly for computing vertical interpolated pixel. And, the average of the horizontal and vertical interpolated pixels is used as the diagonal interpolated pixel.

Given cover image I of size $\frac{M}{2} \times \frac{N}{2}$ is up-sampled to an interpolated image, C of

size $M \times N$ using INP, where C is obtained from I using Eq. (2.1) and (2.3).

$$C(i,j) = \begin{cases} \frac{3C(i,j-1)+C(i,j+1)}{4} & \text{if } i \bmod 2 = 0, j \bmod 2 = 1 \\ \frac{3C(i-1,j)+C(i+1,j)}{4} & \text{if } i \bmod 2 = 1, j \bmod 2 = 0 \\ \frac{C(i-1,j)+C(i,j-1)}{2} & \text{otherwise} \end{cases} \tag{2.3}$$

2.2.4 Enhanced Neighbor Mean Interpolation (ENMI)

Chang *et al*. [48] proposed another improved version of NMI technique, ENMI that computes the interpolated pixels considering its all nearest neighborhood pixels. For computing a horizontal or vertical interpolated pixel, it simply computes the average of two nearest original pixels in horizontal and vertical neighborhoods. For computing a diagonal interpolated pixel, it considers the four nearest original pixels in the diagonal neighborhood. Unlike INP technique, it does not consider any weight for these computation. While NMI uses three nearest pixels for computing a diagonal interpolated pixel, all four original neighborhood pixels are considered in ENMI.

Particularly, up-sampling process is modified to obtain an interpolated image, C of size $M \times N$ from a same size of input image, I as in Eq. (2.4).

$$C(i,j) = \begin{cases} I(i,j) & \text{if } (i,j) \bmod 2 = 0 \\ I(i,j-1) & \text{if } j = N-1 \\ I(i-1,j) & \text{if } i = M-1 \\ \frac{I(i,j-1)+I(i,j+1)}{2} & \text{if } i \bmod 2 = 0, j \bmod 2 = 1 \\ \frac{I(i-1,j)+I(i+1,j)}{2} & \text{if } i \bmod 2 = 1, j \bmod 2 = 0 \\ \frac{I(i-1,j-1)+I(i-1,j+1)+I(i+1,j-1)+I(i+1,j+1)}{4} & \text{otherwise} \end{cases} \tag{2.4}$$

2.2.5 Parabolic Interpolation (PI)

Zhang *et al*. reported parabolic interpolation (PI) for the further improvement of the interpolated image quality in [49]. With a set of three known pixels (*i.e*., original pixels), two unknown pixels are computed as such an original image block of size 1×3 is up-sampled to a block of size 1×5. To obtain an up-sampled pixel, a weighted average of its all possible interpolated values and other nearest neighborhood pixels (that are not considered for interpolation) is used. This computation continues for all overlapping blocks of original pixels with separate consideration of interpolated pixels in the image border.

In general, the interpolated pixels are classified in two kinds: (i) pixels on the image border; (ii) pixels not on the image border. For the second kind of pixels, they can be further classified into three categories in terms of the direction forming by themselves and the known (original) pixels used for interpolation. Clearly, the three directions include the horizontal, vertical and diagonal directions.

Fig.2.2 shows three interpolation patterns along the horizontal, vertical and diagonal directions respectively, where the 'dot' symbol represents the known pixels (original pixels of the input image), and the symbol 'circle' represents the interpolated pixels. The interpolated pixel values are determined by two sides. One is the direct interpolated value calculated by a set of parabolic equations formed using the known pixels; the other is the weighted result of its neighbor pixels already known. Therefore, if the weight of the direct interpolated value is $k(k < 1)$, and the sum of the weights of the neighbor pixels is $(1 - k)$. The weight of each neighbor pixel is determined by the reciprocal of the distance from itself to the interpolated pixel.

For example, to illustrate pixel interpolation along the horizontal direction, as shown in Fig. 2.2, $C(i-1, j)$ is a typical pixel for interpolation, whose adjacent pixels (i.e., $C(i-1, j-1)$ and $C(i-1, j+1)$) along the horizontal direction are al-

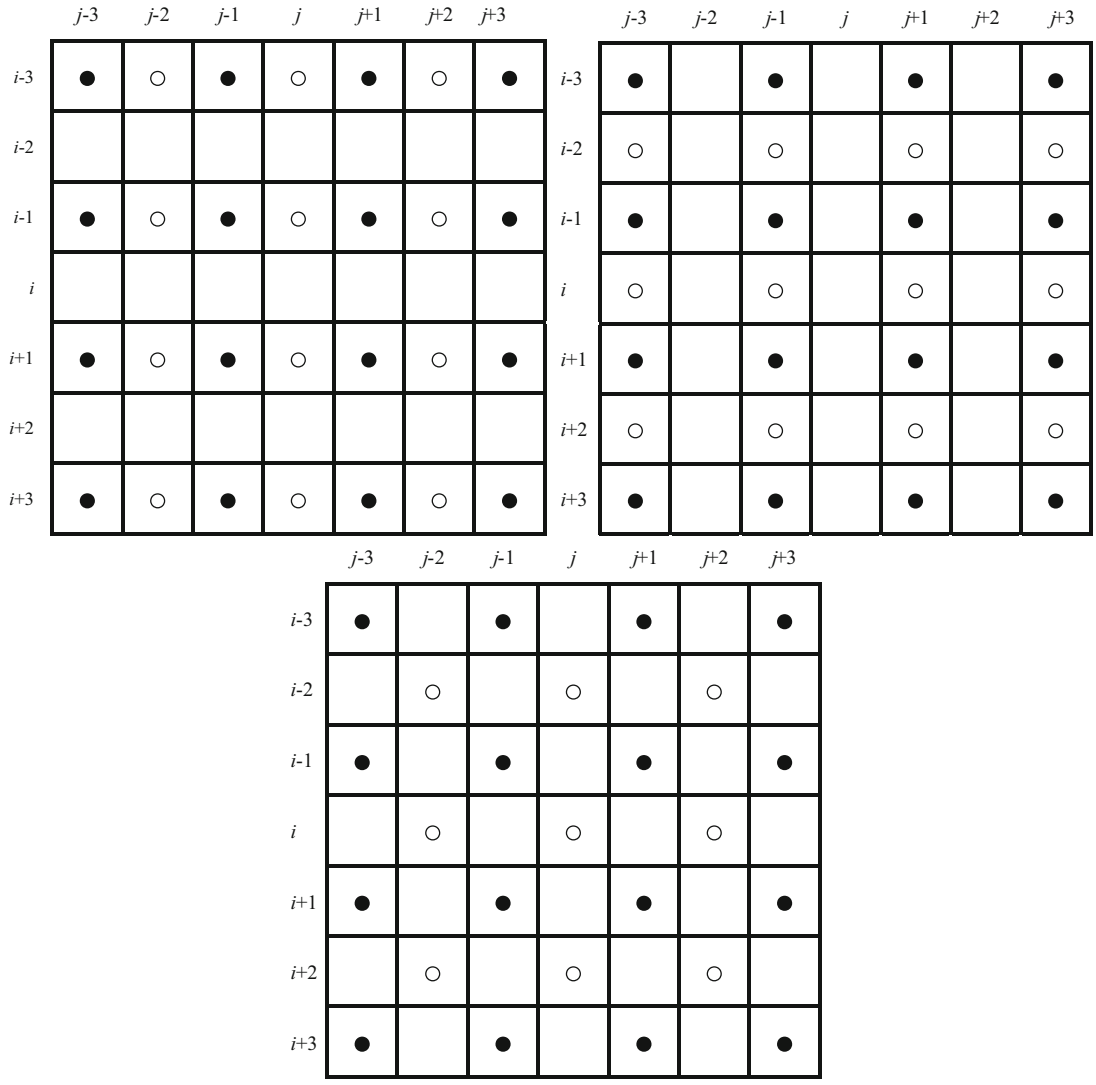


Figure 2.2: Example of computing interpolated pixels using PI along: (a) horizontal, (b) vertical, and (c) diagonal direction.

ready known. In fact, there are both two already known pixels on the left and right sides of $C(i-1,j)$ along the horizontal direction. Therefore, we can obtain two parabolas by different combinations of these known pixels. More specifically, the first parabola is determined by $C(i-1,j-3), C(i-1,j-1), C(i-1,j+1)$, and the second parabola is determined by $C(i-1,j-1), C(i-1,j+1), C(i-1,j+3)$. With the equations determined by the two parabolas, we can calculate two interpolated values at $(i-1,j)$, which are denoted as g_1 and g_2. The general frame-

work of interpolation using the set of parabolic equation is discussed in chapter 3. Moreover, the six known pixels with the smallest distances to the $C(i-1,j)$ are $C(i-1,j-1), C(i-1,j+1), C(i+1,j-1), C(i+1,j+1), C(i-3,j-1)$ and $C(i-3,j+1)$, respectively. Consequently, the $C(i-1,j)$ value is calculated by the weighted results of g_1, g_2, and the six known pixels as follows in Eq. 2.5.

$$C(i-1,j) = k\sum_{i=1}^{2}\frac{g_i}{2} + (1-k)\frac{[C(i-1,j-1)+C(i-1,j+1)+\frac{\sqrt{5}}{5}C(i+1,j-1)+\frac{\sqrt{5}}{5}C(i+1,j+1)+\frac{\sqrt{5}}{5}C(i-3,j-1)+\frac{\sqrt{5}}{5}C(i-3,j+1)]}{\frac{4\sqrt{5}}{5}+2} \tag{2.5}$$

For detailed interpolation of pixels along the other two directions (vertical and diagonal), and pixels on the image border, see [49]. With relatively higher computational complexity, this technique produces better quality interpolated image compared to others which is demonstrated in Table 2.1.

Table 2.1: PSNR Comparison of Different Existing Interpolation Techniques

Test Image	**NNI**	**BI**	**NMI**	**INP**	**ENMI**	**PI**
Airfield	21.208	24.757	25.682	24.716	25.876	26.314
Baboon	20.347	22.32	22.263	21.754	22.721	23.032
Barbara	22.579	24.545	24.499	23.932	24.054	24.935
Boats	23.928	27.51	28.197	27.421	29.198	30.644
Bridge	21.829	24.747	25.109	24.412	25.719	25.775
Couple	23.374	25.896	25.71	24.858	26.582	29.163
Elaine	27.815	31.163	31.03	30.367	31.845	32.116
Goldhill	26.01	29.348	29.877	29.142	30.528	30.69
Lena	25.984	30.382	31.832	30.715	33.493	33.922
Average	**23.675**	**26.741**	**27.133**	**26.369**	**27.780**	**28.510**

Table 2.1 demonstrates the comparison of the existing interpolation techniques discussed above in terms of peak signal to noise ratio (PSNR). It is evident that PI by Zhang *et al*. [49] offers the best possible interpolated image quality so far.

2.3 Embedding Techniques based on Interpolation

We now briefly discuss the use of interpolation techniques for data embedding in IRDH schemes. As mentioned in Chapter 1, the uses of interpolation in IRDH schemes are mainly found for: (*i*) computing prediction errors (PEE-based schemes [40, 41, 53]), (*ii*) computing embeddable bits (DLE-based schemes [25, 45, 46]), and (*iii*) combination of both [48]. An IRDH scheme with PEE computes the predicted errors from the cover image and interpolated image, and the secret data are embedded by modifying the predicted errors in a subset of original pixels. Here, reversibility of these schemes depends on the PEE process. In contrast, an IRDH scheme with DLE embeds secret data only in the interpolated pixels by replacing their LSBs, keeping the original pixels untouched. This embedding is particularly useful in applications like medical and military imaging, where minimum changes in the cover images (*i.e.*, original pixels) are usually restricted [50, 55–59].

We note that in IRDH schemes with PEE, both down-sampling and up-sampling are used successively to retain the size of interpolated and cover images unchanged for computing their predicted errors. Unlike this, only up-sampling is used to compute the interpolated image in IRDH schemes with DLE. So, the cover image can be completely restored by down-sampling the embedded image (*i.e.*, by discarding its interpolated pixels). The interpolated image, if required, can be obtained by re-interpolation of the cover image.

Among the above types of IRDH scheme, DLE-based schemes are relatively simple and offer better user access control. This means that, while the secret data may be extracted by an authorized user (*i.e.*, who has the privilege to obtain the embedded data), the cover image may be independently restored by any user who wants to see it original. For this purpose, a suitable cryptographic tool with private or public keys may be employed for encrypting payload, which is beyond the scope of the thesis. In

summary, IRDH schemes with DLE have several advantages over its counterpart as noted below:

- embedding and extraction processes are relatively simple
- no location map is required
- embedded image being up-sampled provides a higher spatial resolution
- the original pixels remain untouched (required in some military and medical applications)
- instant recovery of the cover image (with or without extraction of embedded data) by down-sampling
- better user access control to the embedded data and cover image, and
- relatively higher embedding capacity with reasonably better image quality.

Considering the above advantages of DLE, we limit our further review in this research, with the development of a DLE-based IRDH scheme. The embedding techniques in some prominent DLE-based IRDH schemes are addressed in the following subsections.

2.3.1 Jung & Yoo's embedding

The interpolated image, C obtained using NMI (see Sec 2.3.1) is utilized for embedding in Jung & Yoo's IRDH scheme. For embedding secret data, the interpolated image C is partitioned into a set of 2×2 non-overlapping blocks. Each block contains one original pixel and three interpolated pixels. In each block, the number of embeddable bits for an interpolated pixel is computed from the difference value between the interpolated and original pixels. Thus, the embeddable data-bits are successively added to an interpolated pixel to obtain respective embedded pixel.

Let the four pixels in a 2×2 non-overlapping block are y_0, y_1, y_2, and y_3, where y_0 is the original pixel and remaining are the interpolated pixels. The number of embeddable

bits n_1, n_2 and n_3 are computed from their difference, d_k for $k \in \{1, 2, 3\}$ using Eq. (2.6a). Now, the set of secret data bits, b_1, b_2 and b_3 of length n_1, n_2 and n_3 are embedded in y_1, y_2 and y_3, respectively to obtain their embedded pixels, y'_k using Eq. (2.6b).

For all 2×2 blocks with $k \in \{1, 2, 3\}$:

$$\left.\begin{aligned} d_k &= y_k - y_0 \\ n_k &= \lfloor \log_2 |d_k| \rfloor \end{aligned}\right\} \tag{2.6a}$$

$$\left.\begin{aligned} y'_k &= y_k + b_k \\ &\text{where } b_k \text{ is the decimal of } n_k \text{ bit data.} \end{aligned}\right\} \tag{2.6b}$$

The Jung & Yoo's IRDH scheme [45] is computationally simple. But, the utilization of the difference between the original pixels and interpolated pixels in a small 2×2 block limits the number of embeddable bits. As a result, the overall embedding rate-distortion performance of the scheme has later been improved with higher block size and better interpolation techniques [46, 48].

2.3.2 Lee & Huang's embedding

Lee & Huang [46] modified the embedding technique of Jung & Yoo for better rate-distortion performance. To increase the number of embeddable bits, a higher difference value is utilized than the Jung & Yoo' scheme. In other words, the difference between an interpolated pixel and the largest original pixel in a 2×2 original pixel neighborhood is used. Finally, the embedded pixel is obtained by embedding the pre-determined number of secret data bits in an interpolated pixel.

Lee & Huang [46] uses INP-based interpolated image for data embedding. Particularly, in Lee & Huang's [46] scheme, for embedding data, the INP-based interpolated

image, C is first partitioned into 3×3 overlapping blocks containing four original pixels in each block to compute block-wise largest original pixel, MV. Then, C is again partitioned into 2×2 non-overlapping blocks such that we get an individual MV for each 2×2 block. Now, for a 2×2 block with the interpolated pixels, $y_k : k \in \{1, 2, 3\}$ and MV, difference values, d_k and corresponding embeddable bits, n_k are calculated using Eq. (2.7). Finally, the decimal values of each n_k-bit of secret data, b_k are embedded in y_k using Eq. (2.6b), and thus the embedded pixels, y'_k are obtained, and so the embedded image I_{em}.

For all 2×2 blocks with $k \in \{1, 2, 3\}$:

$$\left.\begin{aligned} d_k &= MV - y_k \\ n_k &= \lfloor \log_2 |d_k| \rfloor \end{aligned}\right\} \tag{2.7}$$

The above INP-based IRDH scheme utilizes the maximum difference value for embedding and showed better rate-distortion performance over the NMI-based IRDH scheme. Unlike the Jung & Yoo's scheme, consideration of the largest original pixels (out of 4 original pixels) increased the number of embeddable bits in the Lee & Huang's scheme. Thus a higher embedding capacity with reasonably better embedded image quality than the earlier IRDH schemes was demonstrated. However, successive development of IRDH scheme in [48, 49] reported further improved embedding rate-distortion performance.

2.3.3 Chang *et al.*'s embedding

Chang *et al*. [48] later proposed a new IRDH scheme with ENMI technique and two-level embedding. For two-level embedding, Chang *et al.*'s scheme employed combined embedding process of DLE and PEE. Particularly, in the first phase, the number of

embeddable bits are calculated from the difference values between the interpolated pixels and corresponding original pixels. Unlike Jung & Yoo's and Lee & Huang's schemes, the decimal value of the pre-determined set of embeddable data bits is added or subtracted from the interpolated pixel to reduce the embedding distortion. In the second phase, the histogram of the difference image between the embedded image (obtained in the first phase) and original image is modified to obtain the final embedded image with better visual quality.

So, in this scheme, secret data are embedded using the difference of input image, I and interpolated image, C, followed by an HS-based embedding. In the first phase, the number of embeddable bits, n_{em} for the interpolated pixels in C, are calculated from the difference values, D using Eq. (2.8).

$$D(i,j) = C(i,j) - I(i,j) \quad \forall i,j : (i,j) \bmod 2 \neq 0 \tag{2.8a}$$

$$n_{em}(i,j) = \texttt{round}\,(\,log_2\,|D(i,j)|\,) - 1 \tag{2.8b}$$

The embedded image I_{em_1} is obtained by embedding the data bits, d using Eq. (2.9) and (2.10).

$$I_{em_1}(i,j) = C(i,j) \quad \forall\, i,j \;:\; (i,j) \bmod 2 = 0 \tag{2.9}$$

For all i, j such that $(i,j) \bmod 2 \neq 0$:

$$I_{em_1}(i,j) = \begin{cases} C(i,j) + d & \text{if } I(i,j) \geq C(i,j) \\ C(i,j) - d & \text{if } I(i,j) < C(i,j) \\ \text{where } d \text{ is the decimal value of} \\ n_{em}(i,j) - \text{bit data padded with '1' at the LSB.} \end{cases} \tag{2.10}$$

In the second phase, a new difference, D' computed from I_{em} to C and then the histogram of D' is shifted for embedding data in I_{em}. While this second phase of embedding further increases the embedding capacity, it mainly improves the embedded image quality, since the difference of the first phase embedded image is distributed over the second phase embedded image.

Therefore, with ENMI, Chang *et al.*'s scheme achieved a better quality embedded image than the previous schemes. Yet, the use of difference image in the second level of embedding makes the scheme *non-blind*. This means that Chang *et al.*'s scheme requires a set of the original pixels for extracting the data embedded in the first phase of embedding. Additionally, for complete recovery of the original pixels, their original LSBs (substituted with data-bits in embedding) are required to be stored as side information resulting in lower effective embedding capacity.

2.3.4 Zhang *et al.*'s embedding

Zhang *et al.*'s scheme [49] utilized PI for image up-sampling and The relation between the interpolated value and the mean value of a set of original pixels in a particular direction was used to embed secret bits. The secret message is embedded into the interpolated pixels without changing the original pixels. Fig.2.3 shows a 5×5 matrix as a basic unit for data embedding. Fig.2.3 illustrates a basic pattern, where the symbol 'dot' represents the original pixels, and 'circle' represents the interpolated pixels. Total eight sequences $S_i(i = 1, 2, \cdots, 8)$ are found from the matrix according to the directions in each 5×5 block. For each sequence, the $1st$, $3rd$ and $5th$ pixels are the original pixels and their values are kept unchanged, while the remaining two interpolated pixels are modified to embed secret bits. Let the five successive pixels in a sequence be r_1, r_2, r_3, r_4 and r_5. The value of r_2 is y, and the mean value of r_1 and r_3 is u. The difference d is calculated by the equation $d = y - \lfloor u \rfloor$. A large value of $|d|$

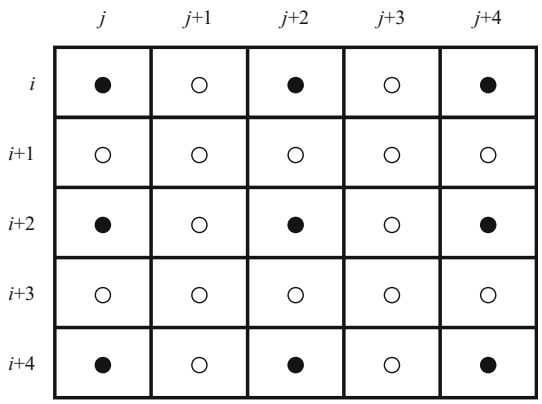


Figure 2.3: Example of the framework of Zhang *et al*. embedding.

means a large change made to the interpolated pixel, which brings a greater distortion to stego-image. In fact, different images have different numbers of pixels with a large $|d|$.

Therefore, to avoid significantly degraded visual quality of the stego-image due to the change of pixels during embedding with a large $|d|$, a threshold is to limit the change range of pixels. Particularly, if $|d|$ is less than the threshold, only then the interpolated pixels would be used for data embedding.

In this case, the number of embedded bits n_{em} can be determined by the equation:

$$n_{em} = \lfloor \log_2 |d| + 1 \rfloor \tag{2.11}$$

The embedding operation is implemented by adding the decimal value of secret bits to the interpolated pixels or subtracting it from the interpolated pixels. The adding or subtracting result either may be closer to the mean $\bar{y_2}$. So, the embedding way is determined by the shape of parabola obtained by the three original pixels. The embedding operation is classified into three cases according to the coefficient, 'a' of the parabolic equation $ax^2 + by + c = y$.

Case 1. If $a < 0$ and the parabola with a downward opening. In this case, data bits are embedded into the interpolated pixel such that, the stego-pixel close to

the mean and not exceed the interpolated value. Suppose that the decimal value of these secret bits is n_1 . The stego-pixel value will be

$$y' = y - n_1 \tag{2.12}$$

Case 2. If $a > 0$ and the parabola with an upward opening, secret bits are embedded into the interpolated pixel. Suppose the decimal value of these secret bits is n_2 . Thus, the stego-pixel value is calculated by Eq. (2.13) .

$$y' = y + n_2 \tag{2.13}$$

Case 3. If $a = 0$ and the parabola degenerates to a line, the number of secret bits embedded into the interpolated pixels is l. Suppose that the decimal value of secret bits is n_3 and the mean value of the nine original pixels in the 5×5 matrix is M. Thus, the stego-pixel value is calculated by Eq. (2.14) .

$$y' = \begin{cases} y + n_3 & \text{if } |y + n_3 + M| \leq |y - n_3 - M| \\ y - n_3 & \text{otherwise} \end{cases} \tag{2.14}$$

Although the consideration of overlapping and directional approach (for both the interpolation and embedding) helps generate a better quality embedded image with higher embedding rate, the scheme eventually remains computationally expensive. Moreover, the above IRDH schemes offer improved interpolated image quality for high embedding capacity, and they do not account for any varying capacity requirement. In summary, there is still need for a more computationally efficient embedding technique of IRDH scheme that offers a better rate-distortion performance particularly for varying embedding capacity requirements.

2.4 Scope of Improvement

The key potentials and limitations of the above prominent IRDH schemes are now summarized below.

The Jung & Yoo's NMI-based IRDH scheme [45] is advantageous with its simple computation and high data embedding rate. But, the number of embeddable bits remains very limited, since it is determined from the difference between the original pixels and interpolated pixels of a small block of size 2×2. That embedding technique offers better embedded image quality, which is further improved later in [46] with the consideration of higher size (3×3) image blocks. Particularly, in the Lee & Huang's INP-based IRDH scheme [46], the maximum difference value for embedding is utilized to demonstrate a better rate-distortion performance over the NMI-based IRDH scheme.

With a more efficient interpolation technique (ENMI), Chang *et al.*'s [48] scheme later achieved a higher image quality than the previous schemes. But, this scheme is *non-blind*, which means that it requires either the difference image $|C - I|$ or the cover image I for extracting the data embedded in the first phase of embedding. Additionally, for complete recovery of the original pixels (which are replaced by the interpolated pixels and used for embedding), their original LSBs are required to be stored as side information resulting in lower effective embedding capacity. In other words, although a combination of both the DLE and PEE in IRDH scheme has also been reported in [48] to offer better image quality and higher embedding capacity, the above mentioned advantages of DLE may no longer exist. For example, the scheme can be semi-reversible and original pixels cannot be preserved intact.

Zhang *et al*. later employed Parabolic interpolation for data embedding. They determined the embedding approach according to the shape of the parabola obtained from three original pixels in different directions. Zhang *et al.*'s scheme [49] improves the

Table 2.2: Summary of Existing DLE-based IRDH Schemes

Schemes	Interpolation	Embedding	Remarks
Jung & Yoo [45]	NMI	Pixel differencing for a 2×2 sub-block	• Interpolation-based reversible data hiding scheme was first proposed. • Simple computation and high embedding rate. • No consideration of varying capacity requirement.
Lee & Huang [46]	INP	Pixel differencing for a 3×3 sub-block	• Improved interpolation and data embedding algorithms. • increased the embedded image quality and embedding capacity. • No consideration of varying capacity requirement.
Chang *et al*. [48]	ENMI	LSB and histogram modification	• Enhanced NMI was used to increase the image quality. • histogram modification was for high embedding capacity. • Non-blind extraction. • No consideration of varying capacity requirement.
Zhang *et al*. [49]	PI	Pixel differencing utilizing the shape of parabola obtained	• Improved interpolated and embedded Image quality. • Computationally costly interpolation. • No consideration of varying capacity requirement.

interpolated and embedded image quality both. However, it seems computationally costly. Table 2.2 represents a summary for a quick review on the existing DLE based IRDH schemes.

Existing IRDH schemes discussed above, although improved the quality of both the interpolated and embedded images with high embedding capacity, they do not account for any varying capacity requirement. Consideration of varying capacity requirement is important, because: for practical applications, payload size usually varies that has remained disregarded. One possible reason for this, it is generally assumed that embedding will continue for the last bit of payload. However, this would result in a low

quality embedded image due to non-uniform distribution of distortion in the image.

This means that, within the maximum embedding capacity limit, embedding of varying size payloads may result in the same level of distortion, which is not desirable and can be effectively managed using a suitably designed adaptive embedding process. So, the current state of the IRDH schemes demands a computationally efficient and suitable adaptive embedding process that offers a better rate-distortion performance. In summary, there is still need for a suitable adaptive embedding process of IRDH scheme that offers a better rate-distortion performance particularly for varying embedding capacity requirements.

Therefore, this research aims at contributing to the development of an adaptive IRDH scheme with more specific focus and plausible outcomes are as follows.

- Use of a better interpolation technique is an important consideration for an IRDH scheme, since the embedded image quality also depends on the interpolated image quality. Zhang *et al.*'s [49] PI technique discussed in Sec. 2.2.5 offers better interpolated image quality, but it is computationally expensive. So, a simplified parabolic interpolation (SPI) technique is developed for efficient up-sampling of the input image with similar or better image quality.
- For varying capacity requirement, a generalized capacity control parameter is formulated and employed in developing a new adaptive embedding process to deterministically allocate the room for embedding. Thus instead of always considering a maximum possible embedding capacity of an image, embedding rate is adaptively managed to attain a given embedding capacity requirement.
- The work in this thesis also aims to offer better embedded image quality. This is ensured with a closer estimate of the embeddable pixel.
- Computational models of all key processes of the proposed IRDH scheme are developed.

- Rate-distortion performance of the proposed IRDH scheme is evaluated, analyzed and validated with the recently reported prominent IRDH schemes.

Thus technical details of the proposed high capacity adaptive IRDH scheme is modeled and discussed in Chapter 3. Experimental results and comparison with existing IRDH schemes are demonstrated in Chapter 4.

2.5 Chapter Summary

A comprehensive detail study of the existing prominent IRDH schemes is carried in this chapter. Different IRDH schemes [45, 46, 48, 49] have contributed for attaining good embedding rate distortion performance. Even though none of the schemes considers varying capacity requirement which leads this research towards the development of an adaptive IRDH scheme which has been presented in the next chapter.

CHAPTER 3

A NEW DATA HIDING SCHEME

3.1 Introduction

All the schemes discussed in the previous chapter offer good image quality and high embedding capacity. However, as mentioned earlier, none of them considers an adaptive approach to attain varying capacity requirement, which is a very significant issue in many applications of RDH schemes.

To attain the varying capacity requirement within the maximum capacity limit, employing an adaptive embedding in IRDH scheme is a solution. So, our research specially focuses on developing a suitable adaptive embedding technique that will account for varying size payloads and ensures uniform distribution of distortion to produce high quality embedded images. With this aim, we Develop an IRDH scheme with a simplified interpolation and adaptive data embedding technique.

This chapter presents the gradual development of our new adaptive IRDH scheme with specific algorithmic and technical details of a computationally efficient simplified interpolation, the adaptive embedding and extraction techniques. We propose a simplified parabolic interpolation (PI) (we, call it SPI) for image up-sampling. For embedding, formal definition of the embedding condition and capacity control parameter are also presented and finally we present our data embedding and extraction techniques. Meaningful examples, necessary explanations, and comparative studies have been presented for supporting the proposed research work.

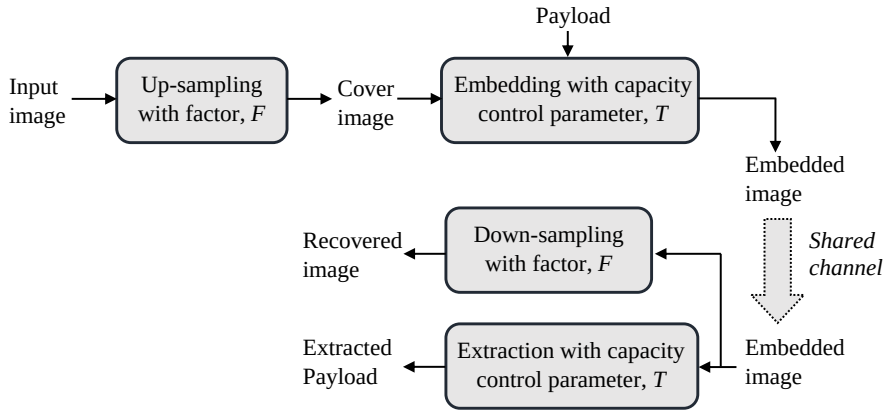


Figure 3.1: General framework of the proposed IRDH scheme

3.2 General Framework of the Proposed Scheme

A general framework of our proposed IRDH scheme is presented in Fig. 3.1. An input image is up-sampled with a suitable factor, the payload is embedded in the up-sampled image (interpolated image) with a formally defined capacity control parameter to obtain the embedded image. The embedded image, while transmitted through a shared communication channel, can be down-sampled to restore the original image and the payload can be extracted using the extraction technique.

Three main processes of the scheme (*i.e.*, image up-sampling, embedding, and extraction) are modelled and discussed with their technical and algorithmic details in sub-sequent sections. Without losing continuity with the notations introduced in Chapter 2, we note that we adopt the general notations of the data hiding framework from [13, 60], which is summarized in Table 3.1. We also differentiate between the embeddable and interpolated pixels as such the embeddable pixels are a sub-set of interpolated pixels, where the payload is embedded in. The other interpolated pixels (*i.e.*, the last three interpolated pixels, see the last paragraph of Sec. 3.4.2) are used to carry any side information required to execute the extraction and recovery processes.

Table 3.1: Key Nomenclature for the Proposed IRDH Framework

Notation	Term
I	input image of size $M \times N \times L$ (*i.e.*, a set of $M \times N$ pixels)
M	number of pixels in a row of I: $M \in \mathbb{N}$
N	number of pixels in a column of I: $N \in \mathbb{N}$
L	bit-depth of an image, $L \in \mathbb{N}$
F	up-sampling factor, $F \in \mathbb{N}$
I_{up}	initialized up-sampled image of size $(FM-1) \times (FN-1) \times L$, *i.e.*, $I_{up} - I = \{0\}$
I'_{up}	up-sampled image of size $(FM-1) \times (FN-1) \times L$ with interpolated pixels
C	interpolated or cover image of size $FM \times FN \times L$
I_{em}	embedded image of size $FM \times FN \times L$
y	an image pixel
n	bit-length of y as in Eq. (3.4)
n_{em}	number of embeddable bits of y as in Eq. (3.3)
y_i, $y_{i,j}$	y of i^{th} or $(i,j)^{th}$ position in 1D or 2D array, respectively, which also applies to y'
y'	an embedded pixel
$data$	set of payload bits, $\{0,1\}^{\gamma}$
γ	embedding capacity requirement (*i.e.*, size of payload in bits, $\gamma = \|\|data\|\|$)
T	capacity control parameter (*i.e.*, number of unchanged MSBs in $y \in C - I$)
I_{Hblock}, I'_{Hblock}	1×5 size image block of horizontal pixels: $I_{Hblock} \subset I_{up}$ and $I'_{Hblock} \subset I'_{up}$
I_{Vblock}, I'_{Vblock}	1×5 size image block of vertical pixels: $I_{Vblock} \subset I_{up}$ and $I'_{Vblock} \subset I'_{up}$
I_{Dblock}, I'_{Dblock}	1×5 size image block of diagonal pixels: $I_{Dblock} \subset I_{up}$ and $I'_{Dblock} \subset I'_{up}$
E_c	total embedding capacity in bits

3.3 Image Up-Sampling

As we discussed in Sec 2.2.5, parabolic interpolation (PI) reported by Zhang *et al*. [49] offers the best possible interpolated image quality so far. Zhang *et al*. used parabolic interpolation in all possible direction in an input image and took a weighted average of the interpolated values in different directions and the neighborhood pixels as the final interpolated value. It results in high computational complexity in terms of runtime (see Sec. 4.3). So, we simplify the PI to reduce the computational complexity with reasonably good image quality. We discuss the general framework of PI first followed by its use in modelling the proposed SPI technique for up-sampling an input image.

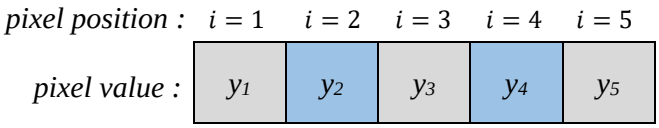


Figure 3.2: Pixel-arrangement for PI.

3.3.1 General framework of PI

A general framework of parabolic interpolation (PI) is illustrated in Fig. 3.2. Let us consider a block of five consecutive pixels, $\{y_i\} : y_i \in \{0, 2^L - 1\}$ with respective pixel positions, $i \in \{1, 2, \cdots, 5\}$, where L is the bit-depth of the given image. With this setting, the known pixels are $\{y_i\} : i \in \{1, 3, 5\}$ and the other two are the up-sampled pixels, *i.e.*, $\{y_i\} : i \in \{2, 4\}$ that are unknown and computed using PI.

$$\left.\begin{aligned} a.1^2 + b.1 + c &= y_1 \\ a.3^2 + b.3 + c &= y_3 \\ a.5^2 + b.5 + c &= y_5 \end{aligned}\right\} \tag{3.1}$$

$$\left.\begin{aligned} y_2 &= a.2^2 + b.2 + c \\ y_4 &= a.4^2 + b.4 + c \end{aligned}\right\} \tag{3.2}$$

For computing the unknown pixels, y_2 and y_4, a set of three parabolic equations of the form $a.i^2 + b.i + c = y_i$ are obtained with the three known pixels as in Eq. (3.1). The coefficients, $a, b,$ and c are then obtained from the solution of these equations, which are used to compute the y_2 and y_4 as in Eq. (3.2).

3.3.2 Proposed SPI technique

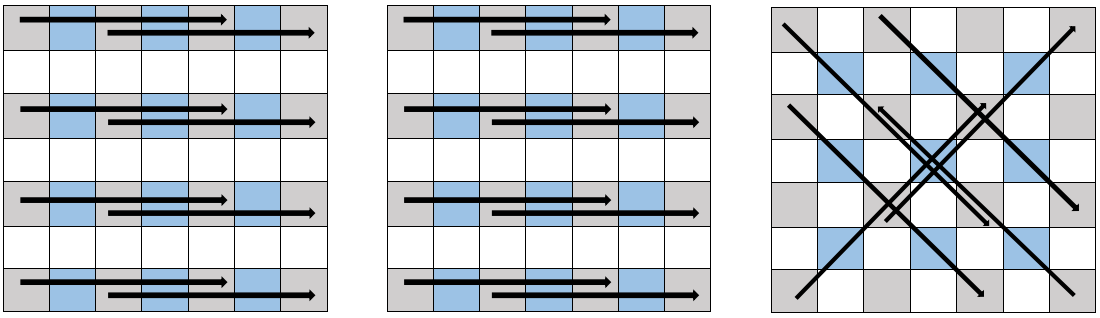
An original image is up-sampled using interpolation with a suitable *scaling factor*. As discussed in Chapter 1, a cover image is required to be up-sampled not only for a higher embedding rate but also for complete and independent recovery (*i.e.*, independent of the extraction techniques) of the cover image.

In our SPI technique, the general computation of PI, discussed above is applied in each directional block (*i.e.*, horizontal, vertical and diagonal) in an image and this process continues for all blocks to compute the complete interpolated image. Our SPI-based up-sampling is modeled as $\texttt{interp}(\cdot)$, key steps of which are explained with important notes in Algorithm 1.

Particularly, $\texttt{interp}(\cdot)$ takes the original image, I and the factor, F as input to output an interpolated image, C. An up-sampled image is initialized with

Algorithm 1: $\texttt{interp}(I, F)$

Input(s): I, F
Output(s): C
Begin
1 $(M, N) \leftarrow \texttt{size}(I)$ ▷ number of rows and columns of I
2 $I_{up} \leftarrow \texttt{upsample}(I, M, N, F)$ ▷ up-sampling I to I_{up} by factor F to a size $FM - 1 \times FN - 1$
3 $\{I_{Hblock}\} \leftarrow \texttt{hbolck}(I_{up})$ ▷ divides I_{up} into a set of overlapping horizontal blocks of size 1×5
4 $\{I_{Vblock}\} \leftarrow \texttt{vbolck}(I_{up})$ ▷ divides I_{up} into a set of overlapping vertical blocks of size 1×5
5 $\{I_{Dblock}\} \leftarrow \texttt{dbolck}(I_{up})$ ▷ divides I_{up} into a set of overlapping diagonal blocks of size 1×5
6 **for all** $I_{Hblock} \subset I_{up}$ **do**
7 $I'_{Hblock} \leftarrow \texttt{parabolic}(I_{Hblock})$ ▷ computes interpolated pixels in I_{Hblock}
8 **end for**
9 **for all** $I_{Vblock} \subset I_{up}$ **do**
10 $I'_{Vblock} \leftarrow \texttt{parabolic}(I_{Vblock})$ ▷ computes interpolated pixels in I_{Vblock}
11 **end for**
12 **for all** $I_{Dblock} \subset I_{up}$ **do**
13 $I'_{Dblock} \leftarrow \texttt{parabolic}(I_{Dblock})$ ▷ computes interpolated pixel sin I_{Dblock}
14 **end for**
15 $I'_{up} \leftarrow \texttt{construct}(\{I'_{Hblock}\}, \{I'_{Vblock}\}, \{I'_{Dblock}\})$ ▷ updates I_{up} with I'_{Hblock}, I'_{Vblock} and I'_{Dblock}
16 $C \leftarrow \texttt{pad}(I'_{up})$ ▷ padding in the last row and last column of I'_{up} to make its size $FM \times FN$
17 Return C
End



(a) (b) (c)

Figure 3.3: Directions in a block for SPI-based image up-sampling: (a) horizontal, (b) vertical and (c) diagonal.

`upsample` $(\cdot)$ that inserts interleaving zero columns and zero rows in I. Thus, an $FM - 1 \times FN - 1$ sized image, I_{up} is computed with the $M \times N$ sized I, which is later divided into a set of overlapping blocks of size 1×5 by scanning pixels in horizontal, vertical and diagonal directions as showed in Fig. 3.3. Thus, as in Step 3-5 of the algorithm, $\{I_{Hblock}\}$, $\{I_{Vblock}\}$ and $\{I_{Dblock}\}$ are the sets of horizontal, vertical and diagonal blocks obtained from the functions, `hblock` $(\cdot)$, `vblock` $(\cdot)$ and `dblock` $(\cdot)$, respectively.

Each directional block contains five consecutive pixels, where the $1st$, $3rd$ and $5th$ pixels are the original, and the $2nd$ and $4th$ pixels are the newly inserted zero pixels, whose values are to be computed using Eq. (3.1) and (3.2) as defined in Steps 9-14 of Algorithm 1 with `parabolic` $(\cdot)$. Newly inserted zero pixel values of $\{I_{Hblock}\}$, $\{I_{Vblock}\}$ and $\{I_{Dblock}\}$ are thus replaced with the interpolated values, which yields their respective interpolated blocks, $\{I'_{Hblock}\}$, $\{I'_{Vblock}\}$ and $\{I'_{Dblock}\}$. These interpolated blocks are then combined to construct I'_{up} using the function, `construct` $(\cdot)$. The last row and last column of I'_{up} is replicated as padded pixels to make the image size $FM \times FN$ using `pad` $(\cdot)$ in Step 16. For example, with $F = 2$, an input image I of size $M \times N$ is finally up-sampled to an interpolated image, C of size $2M \times 2N$.

We note that the value of F is considered here to be a multiple of 2. For any higher

value of F, the up-sampling process with $F = 2$ will thus repeat $\frac{F}{2}$ times (*e.g.*, two times for $F = 4$), which may be needed for any higher capacity requirement as pointed out in Sec. 2.4). For simplicity, we consider here that I is of size 256×256, and with $F = 2$, C would be of size 512×512. Main steps of up-sampling for constructing C from I are again summarized below.

Step 1: A number of empty rows and columns are inserted interleaving the original rows and columns of the input image, I. Thus, a $2M - 1 \times 2N - 1$ sized up-sampled image, C is constructed.

Step 2: The up-sampled image, C is considered as a number of overlapping blocks either in the horizontal, vertical, or diagonal direction at a time. Each block consists of five consecutive pixels among which three are original pixels and the rest two are the up-sampled pixels.

Step 3: Interpolated values of all the up-sampled pixels in each block of C are computed using parabolic interpolation (for the general framework see, Sec. 3.3.1).

Step 4: C is updated with the interpolated pixel values computed in Step 3.

Step 5: Finally, necessary padding is done in the last row and last column of C to make its size $2M \times 2N$. We used padding that replicates the border pixels (*i.e.*, pixels in the $(2M - 1)^{th}$ row and $(2N - 1)^{th}$ column) of C to its $2M^{th}$ row and $2N^{th}$ column and returns the final interpolated image.

3.4 A New Adaptive Embedding

We employ the proposed SPI technique in a new adaptive embedding. In our proposed embedding technique, a pseudo-random set of given data bits $data$ are embedded into the interpolated pixels (we call them *embeddable* pixels) as payload, keeping the origi-

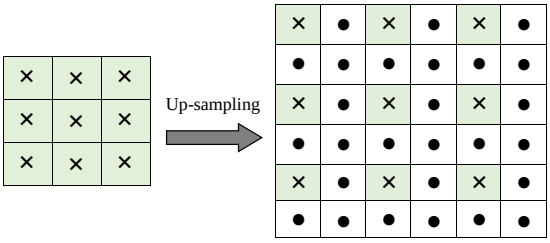


Figure 3.4: Structure of a part of an interpolated image, where *dot* and *cross* represent the up-sampled or interpolated pixels and original pixels, respectively.

nal pixels untouched in an interpolated image, C. For example, see Fig. 3.4, a part of C is shown with the cross-and dot-marks denoting original pixels and embeddable pixels, respectively. Initially, LSBs of the interpolated pixels are investigated to be used for embedding payload either in their original or complement form for minimizing embedding distortion. A flag bit is used to track whether the embedded data is original or its complement required for extracting the payload. Finally, that embedding technique is revised reinvestigating the correlation between the embeddable pixels and the payload bits avoiding the use of any flag bit to ensure significant increment in the embedding capacity further. Thus the main contribution of this thesis *i.e.*, the new adaptive embedding technique is presented in the following two main subsections: (i) adaptive embedding with flag, and (ii) adaptive embedding without flag.

3.4.1 Proposed adaptive embedding with flag

Towards the development of an adaptive embedding technique, a capacity control parameter is employed to allocate the room for embedding. A closer estimate of the embeddable pixels using either the original or the complement of the payload bits is computed to minimize embedding distortion. To track the versions of the embedded bits a flag bit is used.

For a given interpolated (embeddable) pixel, y, number of embeddable bit, n_{em} is determined from the conditions given in Eq. (3.3) based on the bit length, n and the

capacity controlling parameter, T.

$$n_{em} = \begin{cases} 2 & \text{if } n \leq 1+T \\ n-T & \text{otherwise} \end{cases} \tag{3.3}$$

Here, the bit length, n of y is determined from Eq. (3.4). A suitable value of T is selected for different embedding capacity requirement. For an 8-bit image, T remains in the range $[1-6]$.

$$n = \begin{cases} \lceil \log_2(y) \rceil & \text{if } y > 1 \\ 1 & \text{otherwise} \end{cases} \tag{3.4}$$

As in Eq. (3.3), 2-bit data is embedded in the LSBs of the pixel, y if $n \leq 1+T$. This is simply accomplished by replacing the two LSBs of y by the given 2-bit data. For the other case (*i.e.* if $n > 1+T$), the $n-T-1$ bit of data and one flag bit (total $n-T$ bits) are embedded in y. In this case, T-most significant bits (MSBs) of y are kept untouched and next $n-T-1$ bits are used for data and the rest LSB is reserved as a flag bit. The flag bit is used to indicate whether the data is embedded in its original or complement form.

The choice of embedding the data bits either in its original or complement form along with the flag bit is based on the difference between the versions (y_d or $y_{\bar{d}}$) of the embedded pixel, y' and the embeddable pixel, y. Here y_d is the first version of y', which is computed by embedding the $n-T-1$ bit data in y in original, keeping the flag bit 0. Similarly, by embedding $n-T-1$ bit data in its complement form keeping flag bit 1, another version, $y_{\bar{d}}$ is computed. Finally, the embedded pixel, y' takes either the value of y_d or $y_{\bar{d}}$, which is closer to the embeddable pixel, y as in Eq. (3.5). So, the idea of using this flag-bit is to minimize further the distortion caused by the embedding.

$$y' = \begin{cases} y_d & \text{if } |y - y_d| \leq |y - y_{\bar{d}}| \\ y_{\bar{d}} & \text{otherwise} \end{cases} \tag{3.5}$$

Note that the value of T (*e.g.*, 1 to 6) is stored in the LSBs of the first three original pixels and those original LSBs are appended with the data to be embedded for blind extraction of the embedded data. Later the LSBs of these original pixels are readily retrieved from extracted data and these fixed three pixels can be restored.

Algorithm 2 presents our adaptive embedding with flag which is further illustrated. The bit length n of an embeddable pixel $y \in C$ is determined using Eq. (3.4). To start with our embedding technique, as in Algorithm 2, $n - T - 1$ bit (if $n > 1 + T$, otherwise 2-bit) data are extracted from the given $data$ and a flag-bit is appended to it and thus we get $n - T$ bit data (if $n > 1 + T$, otherwise 2-bit data without any flag-bit) to be embedded in y. Before embedding these data-bits, we reset the $n - T$ LSBs of y, and get y_r. Let the decimal value of $n - T$ bit data is d and its complement value is $\bar{d} = 2^{n-T} - d - 1$. Now, for $n > 1 + T$, two versions of embedded pixels y_d and $y_{\bar{d}}$ are computed. The y_d is obtained by embedding d in y, *i.e.* by adding d with y_r. Similarly, $y_{\bar{d}}$ is obtained by embedding $\bar{d}$ in y. The embedding is performed according to the condition given in Eq. (3.5), which implies that the embedded pixel y' finally takes either the value of y_d or $y_{\bar{d}}$, which is closer to the embeddable pixel y. On the other hand, when $n \leq 1 + T$, y_d is obtained by only embedding 2-bit of data without any flag-bit in it. As discussed earlier, the flag-bit (the LSB) in d is kept '0' such that $\bar{d}$ contains the flag-bit 1 to indicate that the d is complemented.

It is noted that for blind extraction, the value of T is also to be stored. It is desirable that only the embedded image should be sufficient to extract the data and to recover the cover image from it. To do this the value of T is stored as a 3-bit data in the LSBs of the first three original pixels of C as pointed at line 29 of Algorithm 2. For

Algorithm 2: `embedding with flag` $(I, data, T, type, F)$

Input(s): I, $data$, T, $type$, F

Output(s): I_{em}

Begin

Initialize:
$startbit \leftarrow 1$
$C \leftarrow \texttt{interp}(I, F, type)$ ▷ up-sampling I to C
$(M, N) \leftarrow \texttt{size}(C)$ ▷ number of rows and columns of C
$temp \leftarrow \texttt{getfixedLSB}(C)$ ▷ returns the LSBs of the first three original pixels of C
$data \leftarrow \texttt{append}(temp, data)$ ▷ 3-LSBs in $temp$ are appended with $data$
for all $y \in C$ **do** ▷ $\{y\}$ are the interpolated pixels in C
  $n \leftarrow \texttt{bitlength}(y)$ ▷ returns bit-length of y using Eq.(3.4)
  if $n \leq 1 + T$ **then**
    $n_{em} \leftarrow 2$ ▷ n_{em} is the number of embeddable bits
    $y_r \leftarrow \texttt{resetLSB}(y, n_{em})$ ▷ resets 2-LSBs of y
    $d \leftarrow \texttt{getbit}(data, n_{em}, startbit)$ ▷ returns decimal value of n_{em}-bits of $data$ from starting bit-position, $startbit$
    $startbit \leftarrow startbit + n_{em}$
    $y' \leftarrow y_r + d$
  else
    $n_{em} \leftarrow n - T$
    $y_r \leftarrow \texttt{resetLSB}(y, n_{em})$ ▷ resets n_{em}-LSBs of y
    $z \leftarrow \texttt{getbit}(data, n_{em} - 1, startbit)$ ▷ returns decimal value of $n_{em} - 1$-bits of $data$ from starting bit-position $startbit$
    $d \leftarrow 2 * z$ ▷ padding a zero flag-bit to the LSB of z
    $\bar{d} \leftarrow \texttt{complement}(d)$ ▷ returns 2's complement of d
    $startbit \leftarrow startbit + n_{em} - 1$
    $y_d \leftarrow y_r + d$
    $y_{\bar{d}} \leftarrow y_r + \bar{d}$
    if $|y - y_d| \leq |y - y_{\bar{d}}|$ **then**
      $y' \leftarrow y_d$
    else
      $y' \leftarrow y_{\bar{d}}$
    end if
  end if
  $y \leftarrow y'$ ▷ updating y with y'
end for
  $I_{em} \leftarrow C$
  $I_{em} \leftarrow \texttt{fixedEMB}(I_{em}, T)$ ▷ value of T is embedded in the LSBs of the first three original pixels of I_{em}
  Return I_{em}

End

example, if $T = 4$, 3 bits of $(4)_{10} = (100)_2$ are stored in LSBs of the three fixed pixels. The original LSBs of those fixed pixels are embedded along with the data. Later, the value of these fixed pixels can be restored by retrieving their original LSBs from the extracted data. The embedded payload (*data*) can be extracted by the reverse process of Algorithm 2.

It will be shown in Chapter 4 that the above embedding technique with flag offers high embedding capacity with better embedded image quality. However, the embedding capacity can further be significantly increased if the use of flag bit is avoided.

So, the thesis continues the investigations for the use of logical (*i.e*. bit-wise) correlation between the embeddable pixels and two versions (*i.e*., original and complement) of embedded pixels. Thereby, the use of flag-bit can be avoided to embed one more payload bit in every place of flag-bits, which would significantly increase the embedding capacity without any additional embedding distortion (see Sec. 3.4.2). Thus a new adaptive embedding technique without flag is developed for varying size payload embedding with the best possible embedded image quality. The following subsection presents the new embedding without flag with algorithmic detail including more formal definition of the capacity control parameter, T for adaptive embedding.

3.4.2 Proposed adaptive embedding without flag

We model the proposed embedding technique here without flag such that it can adaptively embed the payload into the LSBs of the embeddable pixels avoiding the use of any flag bit as mentioned in the previous subsection.

For this, the capacity control parameter, T is defined first in Eq. (3.6) for adaptive embedding, *i.e*., to attain any varying embedding capacity requirement.

$$T \in \{t\} \subset \mathbb{N} : 1 \leq t \leq (L-1) \text{ and } L \text{ is the bit-depth of } C \tag{3.6}$$

In general, T is a predefined number of unchanged MSBs. So, the lesser is the value of T, the greater are the number of embeddable LSBs, and so is the embedding capacity. However, with a higher capacity requirement, embedding distortion is usually higher. So, a maximum possible value of T is required as such the capacity requirement, γ is attained with the best possible embedded image quality. We model the process of computing T in Algorithm 3, where T lies in the range $[1, L-1]$. For different values of T, the embedding capacity, E_c is computed in Step 10 of Algorithm 3 using Eq. (3.7a), which is the total number of embeddable bits, *i.e.*,

Algorithm 3: `c.parameter` $(C, data, F)$

Input(s): C, $data$, F
Output(s): T
Begin
1 **Initialize:**
  $T \leftarrow (L-1)$
  $\gamma \leftarrow$ `size` $(data)$ ▷ γ is the size of payload, $data$
2 $I \leftarrow$ `downsample` (C, F) ▷ down-sample C by F
3 $N_1 \leftarrow 0$ ▷ N_1 is the total pixels with two embeddable LSBs as per Eq. (3.9)
4 **for all** $y \in C - I$ **do** ▷ for all embeddable pixels in C
5 $n \leftarrow$ `bitlength` (y) ▷ returns bit-length of y using Eq. (3.8)
6 **if** $n \leq 1 + T$ **then**
7 $N_1 \leftarrow (N_1 + 1)$
8 **end if**
9 **end for**
10 $E_c \leftarrow N_1 \times 2 + \{(F-1)(M \times N) - N_1\} \times (n - T)$ ▷ E_c is total embedding capacity
11 **while** $E_c \leq \gamma$ **do**
12 **if** $T \geq 2$ **then**
13 $T \leftarrow (T-1)$
14 repeat step 3 to step 10 ▷ recalculate E_c for new T
15 **else**
16 break and reconsider F ▷ Increase F to attain γ
17 **end if**
18 **end while**
19 Return T
End

$\sum n_{em}$ for all $y \in C - I$ as per Eq. (3.7a).

Algorithm 3 illustrates that with the initialized $T = L - 1$ (*i.e.*, the largest value of T), E_c is computed to verify if the embedding capacity requirement, γ is attained. If not, the value of T is further decreased and this process continues until Eq. (3.7b)

Algorithm 4: `embedding without flag` $(F, C, data, T)$

Input(s): C, $data$, F, T
Output(s): I_{em}
Begin
Initialize:
$startbit \leftarrow 1$
$I \leftarrow$ `downsample` (C, F) ▷ down-sample C by F
for all $y \in C - I$ **do** ▷ for all the embeddable pixels in C
$n \leftarrow$ `bitlength` (y) ▷ returns bit-length of y using Eq. (3.8)
if $n \leq 1 + T$ **then**
$n_{em} \leftarrow 2$ ▷ n_{em} is the number of embeddable bits
else
$n_{em} \leftarrow n - T$
end if
$p \leftarrow$ `getLSB` (y, n_{em}) ▷ returns n_{em}-LSBs of y
$d \leftarrow$ `getbit` $(data, n_{em}, startbit)$ ▷ returns n_{em}-bits of $data$ from starting bit-position, $startbit$
$\bar{d} \leftarrow$ `complement` (d) ▷ returns 2's complement of d
$X_d \leftarrow$ `bitXOR` (p, d) ▷ bit-wise XOR of p and d
$X_{\bar{d}} \leftarrow$ `bitXOR` $(p, \bar{d})$ ▷ bit-wise XOR of p and $\bar{d}$
$y_d \leftarrow$ `substituteLSB` (y, d) ▷ returns a version of embedded pixel substituting d for n_{em}-LSBs of y
$y_{\bar{d}} \leftarrow$ `substituteLSB` $(y, \bar{d})$ ▷ returns another version of embedded pixel substituting $\bar{d}$ for n_{em}-LSBs of y
$startbit \leftarrow startbit + n_{em}$
if $|X_d| \leq |X_{\bar{d}}|$ **then**
$y' \leftarrow y_d$
else
$y' \leftarrow y_{\bar{d}}$
end if
$y \leftarrow y'$ ▷ updating y with y'
end for
Return I_{em}
End

is satisfied. It is to be noted that with the lowest value of T (*i.e.*, $T = 1$), the capacity condition in Eq. (3.7b) may not be fulfilled for some large size of payload, γ. This means that the payload size may sometimes exceed the maximum embedding capacity. While this condition may also be true for other RDH schemes, an option to reconsider the up-sampling factor, F to increase the number of embeddable pixels is left here. However, we here illustrate a case of our IRDH scheme with $F = 2$, which may be increased for higher embedding capacity requirement with a higher size interpolated image.

$$E_c = N_1 \times 2 + \{(F-1)(M \times N) - N_1)\} \times (n - T) \tag{3.7a}$$

$$E_c \geq \gamma \tag{3.7b}$$

Once T is obtained for a given capacity requirement, γ, embedding process is invoked for embedding given payload bits. The proposed embedding is modeled in Algorithm 4. The algorithm takes C , $data$, F and T as a set of inputs and returns the embedded image, I_{em}. For example, with a given embeddable pixel $y \in C - I$ the bit length, n is calculated by the function `bitlength` $(\cdot)$ using Eq. (3.8). The number of embeddable LSBs, n_{em} is then computed according to the condition given in Eq. (3.9) by the function `getnem` $(\cdot)$. The number of embeddable LSBs, n_{em} is then computed according to the condition given in Eq. (3.9) by the function `getnem` $(\cdot)$. Now, n_{em}-number of LSBs of y is compared with the n_{em}-bit $data$, d and its complement, $\bar{d}$ using bit-wise XOR operation. Thus, their logical differences, X_d and $X_{\bar{d}}$ are computed, respectively. These X_d and $X_{\bar{d}}$ are compared according to the condition given in Eq. (3.10) to choose the final embedded pixel, y' from y_d and $y_{\bar{d}}$. (Here, y_d and $y_{\bar{d}}$ are the two versions of embedded pixel computed using `substituteLSB` $(\cdot)$; the first version replaces the LSBs of y with the d, and the second one replaces the same with $\bar{d}$.) In other words, $|X_d| \leq |X_{\bar{d}}|$ means that embeddable pixel version, y_d is closer to y. So, the embedded pixel, y' would take the value of y_d, otherwise the

embedded pixel would be $y_{\bar{d}}$. Continuing this embedding for all embeddable pixels, an embedded image, I_{em} is obtained.

$$n = \begin{cases} \lceil \log_2 (y) \rceil & \text{if } y > 1 \\ 1 & \text{otherwise} \end{cases} \tag{3.8}$$

$$n_{em} = \begin{cases} 2 & \text{if } n \leq 1 + T \\ n - T & \text{otherwise} \end{cases} \tag{3.9}$$

$$y' = \begin{cases} y_d & \text{if } |X_d| \leq |X_{\bar{d}}| \\ y_{\bar{d}} & \text{otherwise} \end{cases} \tag{3.10}$$

However, for a blind operation of the recovery and extraction processes, the value of both F and T are to be stored as side information in the I_{em} such that the cover image can completely be recovered and the embedded data are exactly extracted from I_{em}. For this, the last three interpolated pixels are excluded for embedding. In each of these pixels, two bits of the 6-bit side information (the first 3-bits are for F and the rest 3-bits are for T) are embedded so that the receiver can extract these bits to execute the extraction and recovery processes. The other interpolated pixels used for payload embedding are defined as the embeddable pixels in Algorithm 4 and 5. So, without loss of generality, we omit the embedding and extraction of F and T in the algorithms assuming that this pre-processing can be employed later for a practical application scenario.

3.5 Data Extraction with Image Recovery

Once the embedded image is sent to the receiver as showed in Fig. 3.1, extraction of the embedded payload bits and recovery of the original image take place. For both the embedding techniques (with and without flag) presented in Sec. 3.4.1 and Sec. 3.4.2, the extraction process is reversible. Since the embedding without flag offers higher capacity with better image quality (see Sec. 4.4), without losing the generality, data extraction with image recovery of the proposed IRDH scheme considering the embedding without flag is discussed in this section only. It is modeled in Algorithm 5. The original image, I can be restored instantly by down-sampling the embedded image, I_{em}, *i.e.*, discarding the embedded pixels and the embedded payload can be blindly extracted from the I_{em}. Once the original image, I is restored, the interpolated image, C is reconstructed using `interp`$(\cdot)$. Now, for all embedded pixels $y' \in I_{em} - I$, the bit length, n of y' and the number of embedded bits, n_{em} in y' are calculated with `bitlength`$(\cdot)$ and `getnem`$(\cdot)$ using Eq. (3.11) and (3.9), respectively .

$$n = \begin{cases} \lceil \log_2 (y') \rceil & \text{if } y' > 1 \\ 1 & \text{otherwise} \end{cases} \tag{3.11}$$

$$d = \begin{cases} b & \text{if } |X_b| \leq |X_{\bar{b}}| \\ \bar{b} & \text{otherwise} \end{cases} \tag{3.12}$$

Now, n_{em}-number of LSBs of $y' \in I_{em} - I$ and corresponding $y \in C - I$ are extracted as b and p, respectively using `getLSB`$(\cdot)$. Then b and p are compared using bit-wise XOR operation to determine if the set of embedded bits is equal to b or its complement, $\bar{b}$ as per Eq. (3.12). These operations are stated in Steps 13-22 of Algorithm 5. Therefore, all the extracted payload-bits, d are concatenated by `append`$(\cdot)$ to reconstruct the extracted payload, D (which is the same as the embedded payload, $data$).

Algorithm 5: `extraction` (I_{em}, F, T)

Input(s): I_{em}, F, T
Output(s): D, I
Begin
Initialize:
    $D \leftarrow Null$ ▷ D is an empty array to store $data$
$I \leftarrow$ `downsample` (I_{em}, F) ▷ down-sample I_{em} by F
$C \leftarrow$ `interp` (I, F) ▷ up-sampling I to C by factor F
$\{y'\} \leftarrow I_{em} - I$ ▷ $\{y'\}$ are the set of embedded pixels in I_{em}
$\{y\} \leftarrow C - I$ ▷ $\{y\}$ are the set of embeddable pixels in C
for all y' and y **do**
    $n \leftarrow$ `bitlength` (y') ▷ returns bit-length of y' using Eq.(3.11)
    if $n \leq 1 + T$ **then**
        $n_{em} \leftarrow 2$ ▷ n_{em} is the number of embeddable bits
    else
        $n_{em} \leftarrow n - T$
    end if
    $p \leftarrow$ `getLSB` (y, n_{em}) ▷ returns n_{em}-LSBs of y
    $b \leftarrow$ `getLSB` (y', n_{em}) ▷ returns n_{em}-LSBs of y'
    $\bar{b} \leftarrow$ `complement` (b) ▷ returns 2's complement of b
    $X_b \leftarrow$ `bitXOR` (p, b) ▷ bit-wise XOR of p and b
    $X_{\bar{b}} \leftarrow$ `bitXOR` $(p, \bar{b})$ ▷ bit-wise XOR of p and $\bar{b}$
    if $|X_b| \leq |X_{\bar{b}}|$ **then**
        $d \leftarrow b$ ▷ data bits-are embedded in original
    else
        $d \leftarrow \bar{b}$ ▷ data bits-are embedded in complement
    end if
    $D \leftarrow$ `append` (D, b) ▷ append selected data bits, b with extracted Data, D
end for
Return I, D
End

3.6 A Working Example

We now illustrate the working of the two embedding techniques with minimal example below. Firstly, we will see an example of embedding payload into an interpolated image with flag only. Later we will explain the IRDH scheme embedding without flag, which includes illustration of the SPI technique for image up-sampling and pixel-wise

2	4	12	39	123
50	27	29	34	150
195	167	83	120	231
174	170	108	114	222
112	130	185	187	195

(a)

2	7	12	38	123
50	24	28	32	152
195	172	83	120	231
170	172	110	116	216
112	136	185	186	195

(b)

Figure 3.5: A minimal example of interpolation and proposed embedding (with flag): (a) interpolated version and (b) its embedded version with flag.

embedding without flag.

(1) **Embedding with flag:** Fig. 3.5 (a) shows an interpolated image. Let a pseudo-random data (*e.g.*, $data = [1111000001001 1000101110111010010010101111110001111010001111100000010011000 \cdots]$) are embedded into the interpolated pixels to obtain the embedded image as in Fig. 3.5 (b), as discussed in Sec. 3.4.1 according to Algorithm 2.

Specifically, the first embeddable pixel $y = 4$ has bit length $n = 3$, so, considering $T = 4$, $n_{em} = 2$ as it is a case of $n \leq (1 + T)$. This means that, the first 2-bit of the given data $d = (11)_2$ are embedded in y and respective embedded pixel is $y' = 7$ (see Fig. 3.6).

On the other hand, the next embeddable pixel $y = 39$ having bit length $n = 6$, is a case of $n > (1 + T)$. So, with 1-bit (*i.e.*, $(n - T - 1) = 1$) from the data (after the last embedded bit position) and 1-bit flag, two versions of embedded pixels $y_d = 38$ and $y_{\bar{d}} = 37$ are computed. Since $y_d = 38$ is closer to $y = 39$, so the embedded pixel becomes $y' = y_d = 38$ (see Fig. 3.7).

(2) **IRDH Scheme with adaptive embedding (without flag):** We now briefly explain a minimal working example of the proposed IRDH scheme (with SPI and embedding without flag). As illustrated in Fig. 3.8, an input image of size 3×3 is up-sampled

Condition	Fixed MSB	Embeddable bits		Decimal Value
$y = 4 \rightarrow n = 3$, $T = 4 \rightarrow n \leq 1+T$ No. of embeddable bits = 2	1	0	0	= 4
Reset 2 LSBs of y (y_r)	1	0	0	= 4
2 bit of *data* (d)		1	1	= 3
Embedded Pixel value (y')	4 + 3 = 7			= 7

Figure 3.6: Example of proposed embedding with flag in a given pixel : $n \leq 1 + T$

Condition	Fixed MSB	Embeddable bits	Flag	Decimal Value
$y = 39 \rightarrow n = 6$, $T = 4 \rightarrow n > 1+T$ Embeddable data: (n-T-1 = 1) bit *data* and 1 bit flag	1 0 0 1	1	1	= 39
Reset 2 LSBs of y (y_r)	1 0 0 1	0	0	= 36
1 bit data		1		
data with flag bit (d)		1	0	= 2
Compliment of d ($\bar{d}$)		0	1	= 1
y_d	(36 + 2 = 38)			= 38
$y_{\bar{d}}$	(36 + 1 = 37)			= 37
Embedded Pixel value (y')	Since y_d is closer to y			= 38

Figure 3.7: Example of proposed embedding with flag in a given pixel: $n > 1 + T$

to an image of size 6×6 with $F = 2$ and the interpolated pixels are calculated using our SPI technique (Algorithm 1). The up-sampling of the input image is initialized with the interleaving zero columns and zero rows as in Fig 3.8(b). The darker pixels in the figure represent the original pixels and these pixels are kept unchanged. A directional block of five pixels is considered in all possible horizontal, vertical and diagonal directions to compute two unknown pixels from three known pixels as mentioned in Section 3.3.2.

For a minimal working example, we explain here the first horizontal block, where the three known pixels are $y_1 = 152$, $y_3 = 161$ and $y_5 = 192$ and the unknown pixels are y_2 and y_4. We initialize the up-sampling process for the unknown pixels, y_2 and y_4 with zeros, which are to be replaced by the respective interpolated values. We derive the set of three parabolic equations as in Eq. (3.1) for the known three pixels, and solve those equations to obtain the coefficients, $a = 2.75$, $b = -6.5$ and $c = 155.75$. With the values of the coefficients, we now compute the unknown pixel values using

Eq. (3.2), *i.e.*, $y_2 = 153.75 = 154$ and $y_4 = 173.75 = 174$, where the values are rounded up to the nearest integer numbers. This process of obtaining two interpolated pixels from three known pixels continues for all the directional blocks to obtain the final interpolated image with necessary padding as illustrated in Fig 3.8(c).

We now illustrate the process of the proposed adaptive embedding without flag. The payload, a pseudo random binary data (*e.g.*, 11011001110110100111101111101010000110100011000111011000101010111110001010100010010...) is embedded into the embeddable pixels (white colored cells in the figure) and the embedded image is presented in Fig.3.8(c). The embedding process, for example, is now illustrated with $T = 5$ in Fig. 3.9.

Let us consider the first embeddable pixel, $y = (154)_{10} = (10011010)_2$. The bit-length of y is $n = 8$, thus from Eq. 3.9, we get the number of embeddable bits, n_{em} = $n - T$ = (8-5) = 3. The set of the three embeddable LSBs of y is $p = 010$. On the other hand, 3 bit of payload, $d = 110$. This set of payload bits, d is embedded in the embeddable LSBs of y, either in its original form (*i.e.* $d = 110$) or in complement form (*i.e.* $\bar{d}$ = 001). Now, the two versions of embedded pixel, y_d and $y_{\bar{d}}$ can be obtained by substituting d or $\bar{d}$ respectively, for the set of embeddable LSBs, p. To select the final embedded pixel, we utilize bit-wise correlation of p and d. We then compute the value, $X_d = 4$ using bit-wise XOR operation of p and d, and $X_{\bar{d}} = 3$ using bit-wise XOR of p and $\bar{d}$. According to Eq. 3.10, since $|X_d| > |X_{\bar{d}}$, replacement of the LSBs, $p = 010$ with the complement of data bits, $\bar{d} = 001$ (*i.e.*, $y_{\bar{d}}$ version of the embedded pixel) would cause minimum embedding distortion to the embeddable pixel, y. So, we obtain the final embedded pixel, $y' = y_{\bar{d}} = (10011001)_2 = (153)_{10}$.

On the other hand, in extraction, the original image can be exactly recovered by simply discarding the embedded pixels from the embedded image. Applying the reverse process of embedding, we can also completely extract the embedded payload from the embedded pixels as explained in Sec. 3.5.

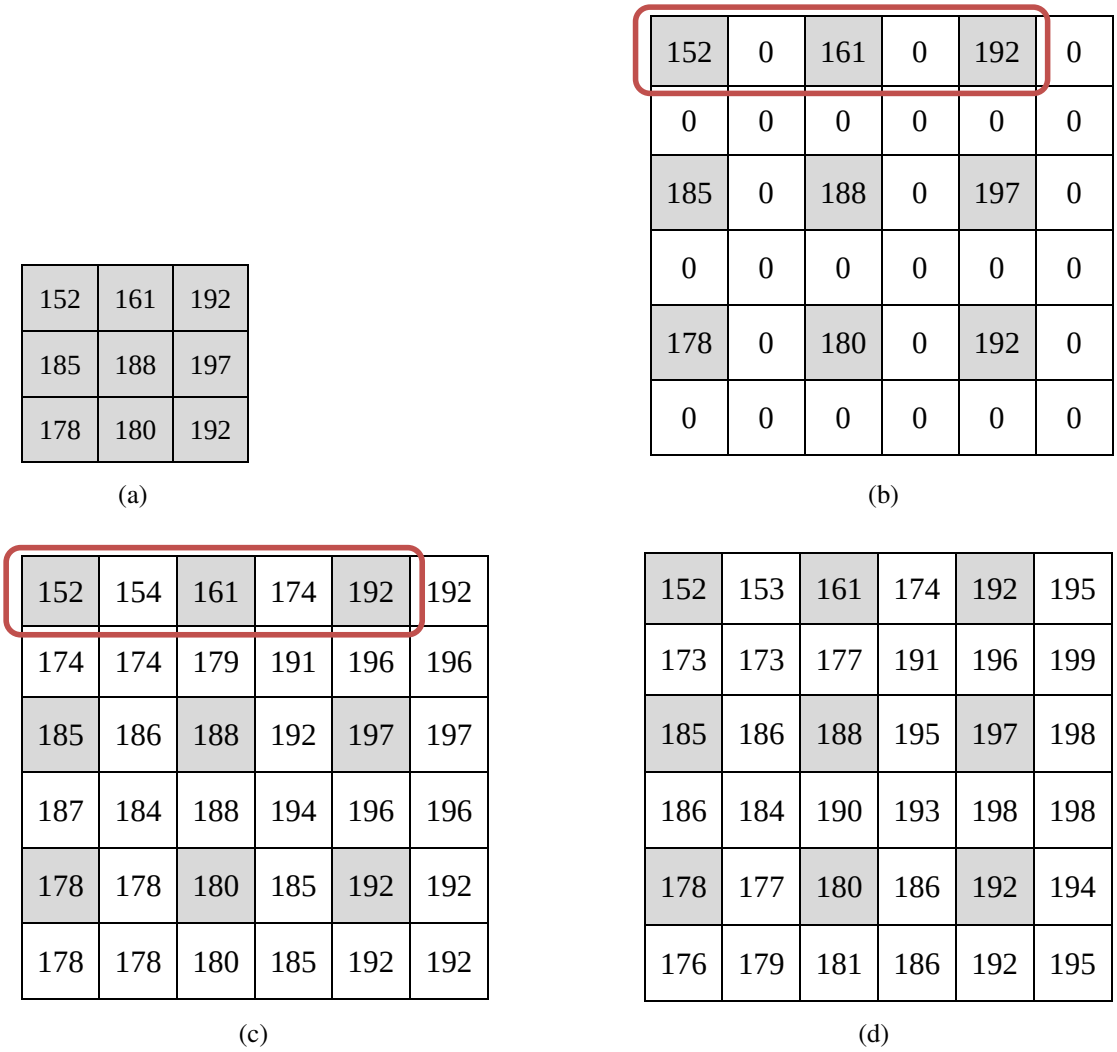


Figure 3.8: A minimal example of the proposed interpolation and embedding processes (without flag): (a) input image, (b) initial up-sampled image, (c) interpolated image, and (d) embedded image (the darker cells represent the original pixels).

3.7 Chapter Summary

This chapter presents the technical details of the new IRDH scheme for varying size payload embedding with necessary working example. The proposed scheme utilizes the SPI technique for image up-sampling and adaptive embedding for data embedding. The embedding techniques embed the data bits either in their original form or complement form to keep the embedding distortion minimum. The performance of the new

Case	Fixed MSB					Embeddable bits			Decimal Value
$y = 154 \rightarrow n = 8$, $T = 5 \rightarrow n > 1 + T$ ∴ no of embeddable bits, $n_{em=}\ n - T = 3$	1	0	0	1	1	0	1	0	**154**
$n_{em}(= 3)$ LSBs of $y = p$						0	1	0	
$n_{em}(= 3)$ bit data $= d$						1	1	0	
Complement of $d = \bar{d}$						0	0	1	
y embedded with $d = y_d$	1	0	0	1	1	1	1	0	158
y embedded with $\bar{d} = y_{\bar{d}}$	1	0	0	1	1	0	0	1	153
$X_d = bitxor(p, d)$						1	0	0	4
$X_{\bar{d}} = bitxor(p, \bar{d})$						0	1	1	3
∵ $\lvert X_d \rvert > \lvert X_{\bar{d}} \rvert$ ∴ Embedded pixel value $y' = y_{\bar{d}}$	1	0	0	1	1	0	0	1	**153**

Figure 3.9: An example of proposed embedding (without flag) in an embeddable pixel.

adaptive IRDH scheme (SPI-based embedding with and without flag) proposed in this chapter is carried out in the next chapter.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter presents the evaluation of the new IRDH scheme presented in Chapter 3 for its efficiency and better embedding rate -distortion performance. The performance of the proposed scheme is validated with high capacity and better image quality compared to the prominent IRDH schemes. In Sec. 4.3, we analyze the computational efficiency of the proposed SPI technique, and in Sec. 4.4, the embedding rate-distortion performance of our IRDH scheme with and without flag are analyzed and validated by comparing it with some prominent IRDH schemes [45, 46, 48, 49]. The experiment is carried out for a set of standard USC-SIPI gray scale test-images [61] of size 256×256. As mentioned in Sec. 3.3.2, we have used a factor, $F = 2$, for up-sampling the original test images to a size of 512×512.

The proposed IRDH scheme and the existing IRDH schemes [45, 46, 48, 49] were implemented using MATLAB R2016b with a 1.3 GHz Intel core i5 CPU and 4 GB memory. The proposed scheme developed in this research can also be re-implemented with the technical details given in Sec. 3.3 and Sec. 3.4. However, it is to note that, with the programming skills and platforms, the optimization of the implemented code may vary without losing the embedding rate-distortion performance reported in this thesis.

4.2 Evaluation Metrics

The embedding rate-distortion performance of our scheme is evaluated and compared with the existing prominent IRDH schemes [45, 46, 48, 49]. The visual quality of the interpolated and embedded images are evaluated in terms of PSNR (peak signal to noise ratio) in dB and SSIM (structural similarity), two popular image quality metrics. Embedding rate and capacity are expressed in terms of bpp (bit per pixel) and the total number of embedded bits, respectively.

$$\text{MSE} = \frac{\sum_{j=1}^{FN}\sum_{i=1}^{FM}(y_{i,j} - y'_{i,j})^2}{FM \times FN} \tag{4.1}$$

$$\text{PSNR} = 10\log\frac{(2^L-1)^2}{\text{MSE}} \tag{4.2}$$

$$\text{SSIM} = \frac{(2\mu_y\mu_{y'} + c_1)(2\sigma_{y,y'} + c_2)}{(\mu_y^2 + \mu_{y'}^2 + c_1)(\sigma_y^2 + \sigma_{y'}^2 + c_2)} \tag{4.3}$$

The PSNR values for the embedded images are computed in terms of mean square error (MSE) as given in Eq. (4.1) and (4.2). Here, $y_{i,j}$ and $y'_{i,j}$ are the pixel values of position (i, j) in an interpolated image and its embedded version, both of size $FM \times FN$, respectively. Besides, SSIM values are computed using Eq. (4.3), where μ_y and $\mu_{y'}$ are the average values of $y_{i,j}$ and $y'_{i,j}$, and σ_y^2 and $\sigma_{y'}^2$ are the variance of $y_{i,j}$ and $y'_{i,j}$, respectively; $\sigma_{y,y'}$ is the covariance of $y_{i,j}$ and $y'_{i,j}$; and $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ are two regularization constants for the $2^L - 1$ dynamic range of the pixel values and a set of small constants, k_1 and k_2 [62]. Here, L is the bit-depth of the images.

4.3 Computational Efficiency of the SPI Technique

As discussed in Sec. 2.2.5, parabolic interpolation (PI) reported by Zhang *et al*. [49] offers better interpolated image quality among the existing techniques including BI, NNI, NMI, INP, ENMI and PI. However, as PI technique is computationally costly, the

SPI technique is proposed in Sec.3.3.2. Table 4.1 demonstrates the PSNR comparison of the SPI technique with other interpolation techniques for different USC-SIPI test images. As in Table 4.1, the interpolation PSNR for the proposed SPI is 27.159 dB in average which is slight less than the PI technique (28.51 dB). With this little compromise in PSNR, the SPI technique offers significantly less computational time than the PI.

The computational time and efficiency of the SPI technique is compared with Zhang *et al.*'s [49] PI technique in Table 4.2 in terms of their up-sampling run-time. As demonstrated in the Table 4.2, for different test images, the average run-time for the proposed SPI is only 1.81 sec, while that for Zhang *et al.*'s PI [49] is 10.68 sec. This means that, although the PSNR values gets slightly reduced (4.7% in average) with SPI compared to PI, the SPI takes significantly less computational time (83% less in average) to up-sample an input image. Thus computational complexity is significantly reduced with the SPI technique. Thereby, the overall embedding time of the proposed scheme is also significantly reduced. While the PSNR values for the interpolated images are slightly lower, the embedded image quality obtained with our proposed IRDH scheme eventually outperform the Zhang *et al.*'s scheme, which is analyzed in the following sub-section.

Table 4.1: PSNR Comparison of SPI with Other Interpolation Techniques

Test Image	NNI	BI	NMI	INP	ENMI	PI	SPI (Ours)
Airfield	21.208	24.757	25.682	24.716	25.876	26.314	26.117
Baboon	20.347	22.32	22.263	21.754	22.721	23.032	21.893
Barbara	22.579	24.545	24.499	23.932	24.054	24.935	23.585
Boats	23.928	27.51	28.197	27.421	29.198	30.644	28.429
Bridge	21.829	24.747	25.109	24.412	25.719	25.775	24.856
Couple	23.374	25.896	25.71	24.858	26.582	29.163	26.507
Elaine	27.815	31.163	31.03	30.367	31.845	32.116	31.030
Goldhill	26.01	29.348	29.877	29.142	30.528	30.69	29.729
Lena	25.984	30.382	31.832	30.715	33.493	33.922	32.288
Average	**23.675**	**26.741**	**27.133**	**26.369**	**27.780**	**28.510**	**27.159**

Table 4.2: Performance of the PI Techniques

Test Image	PSNR (dB)		Run-time (sec)		
	PI [49]	**SPI (ours)**	**PI [49]**	**SPI (ours)**	**Efficiency (%)**
Airfield	26.31	26.12	11.14	1.89	83.06
Baboon	23.03	21.89	10.04	1.79	82.19
Barbara	24.94	23.58	10.44	1.82	82.56
Boat	30.64	28.43	10.42	1.88	82.00
Bridge	25.78	24.86	10.48	1.82	82.63
Couple	29.16	26.51	10.20	1.79	82.49
Elaine	32.12	31.03	10.26	1.81	82.40
Goldhill	30.69	29.73	10.35	1.76	82.99
Lena	33.92	32.29	12.04	1.77	85.31
Average	**28.51**	**27.16**	**10.68**	**1.81**	**83.07**

We here also note that, if there is no requirement of computational efficiency, the Zhang *et al.*'s [49] PI technique may be used for computing up-sampled images in the proposed IRDH scheme. This would also improve the embedded image quality of the scheme further.

4.4 Rate-Distortion Performance of the Proposed Scheme

For all the schemes including the one developed and presented in this thesis, the embedded image quality is evaluated with respect to the interpolated image in terms of PSNR and SSIM. Embedding rate versus both the PSNR and SSIM values are then compared and analyzed for different values of the capacity controlling parameter, T. Thereby, the efficiency of the proposed scheme is demonstrated for better rate-distortion performance.

We observe that the embedded images by the proposed scheme remains visually imperceptible compared to the interpolated images. Fig. 4.1 and 4.2 demonstrate the visual quality of interpolated and embedded versions of different test images for

different values of T with the proposed scheme with flag and without flag, respectively. Although this comparison is shown there only for a few sample test images, similar visual quality is also obtained for the other test images. A quantitative illustration of the rate-distortion performance is presented in Table 4.3 and 4.4 in terms of PSNR (in dB), SSIM and embedding capacity (in total bits and bpp).

As illustrated in Table. 4.3 and 4.4, the proposed scheme with new adaptive embedding techniques (with and without flag) offers relatively better embedded image quality (*i.e.*, higher values of PSNR and SSIM) that gradually increases with the lower embedding rates and higher values of T. This is because, as mentioned in Sec. 3.4.2, number of embeddable bits decreases (and respective embedded image quality improves), while T increases. Thus, a trade-off between the image quality and embedding rate are made with adaptively chosen value of T, as explained in Algorithm 3, to obtain the best possible image quality for a given capacity requirement. This holds true for both the embedding with or without flag. Moreover, for similar embedding rates with a suitable of value of T, proposed schemes embedded with or without flag, both are also observed to generate a better quality embedded image compared to that obtained by the schemes in [45, 46, 48, 49].

Now let us analyze the performance of the proposed schemes with the existing schemes in the following three subsections: (i) performance of the proposed scheme embedded with flag (ii) performance of the proposed scheme embedded without flag (iii) comparison between the proposed schemes embedded with and without flag.

4.4.1 Performance of the proposed scheme embedded with flag

While compared to the other schemes, the proposed scheme embedded with flag offers better rate-distortion performance for the test images. It can be noted that, unlike the proposed scheme, the other schemes have no consideration of any varying capac-

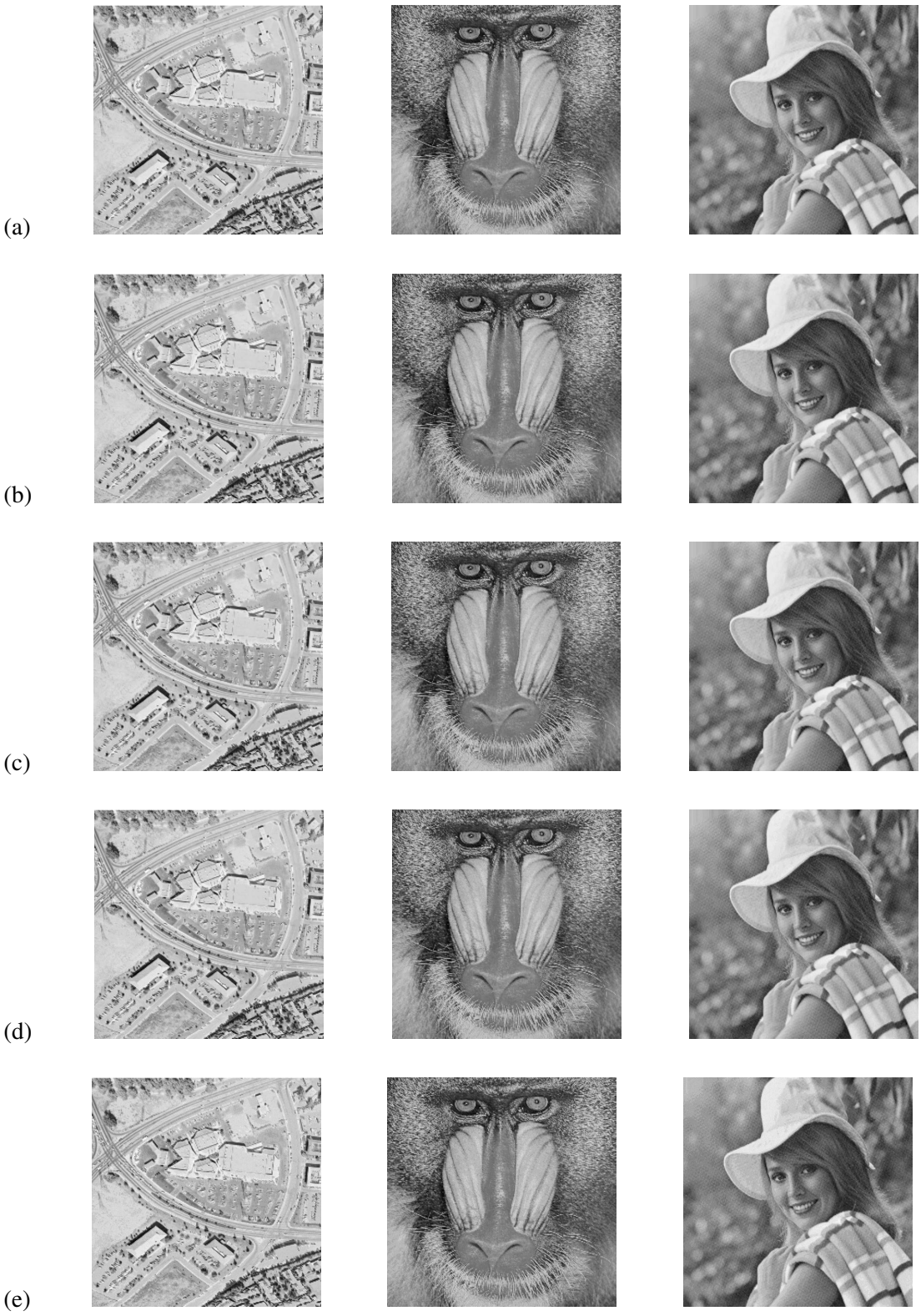


Figure 4.1: Example of output images for proposed scheme with flag: (a) interpolated images, (b) embedded images for $T = 6$, (c) embedded images for $T = 5$, (d) embedded images for $T = 4$ and (e) embedded images for $T = 3$. (Images in each row, from left: Airfield, Baboon and Elaine)

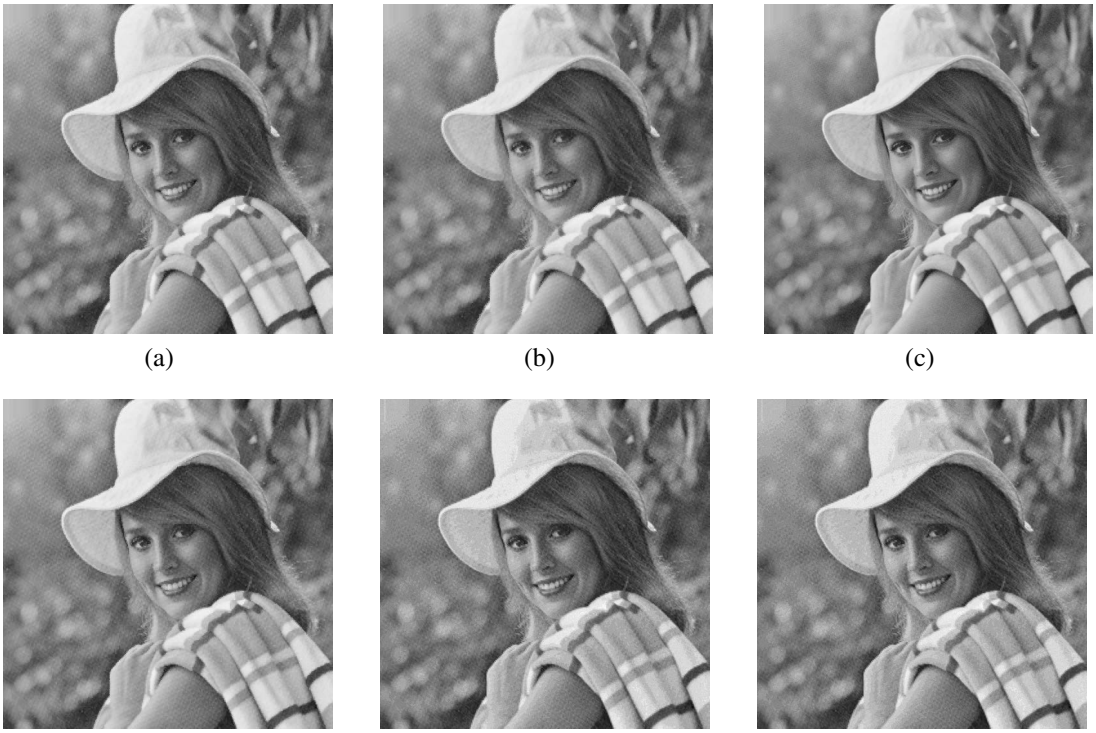
Figure 4.2: Example of output images for proposed scheme without flag: (a) interpolated images, (b) embedded images for $T = 6$, (c) embedded images for $T = 5$, (d) embedded images for $T = 4$ and (e) embedded images for $T = 3$. (Images in each row, from left: Boat, Goldhill and Lena)

Table 4.3: Performance Comparison of the Proposed Scheme

Test Images	Performance Metric	Jung & Yoo [45]	Lee & Huang [46]	Chang *et al.* [48]	Zhang *et al.* [49]	embedding with Flag (Proposed)				embedding without Flag (Proposed)			
						T=3	T=4	T=5	T=6	T=3	T=4	T=5	T=6
Airfield	Total bits	529095	511793	332918	698902	763301	567824	378693	215178	959807	763816	571255	393217
	bpp	2.018	1.953	1.270	2.666	2.912	2.166	1.445	0.821	3.661	2.914	2.179	1.500
	PSNR (dB)	23.85	23.76	28.64	22.53	27.61	33.50	39.55	45.42	33.46	39.53	45.67	50.92
	SSIM	0.8042	0.7970	0.9045	0.8530	0.6494	0.8124	0.9140	0.9707	0.8120	0.9131	0.9691	0.9920
Baboon	Total bits	624709	637491	447173	639317	676493	482541	311718	291703	872940	677741	494730	393217
	bpp	2.383	2.432	1.706	2.439	2.581	1.841	1.189	1.113	3.330	2.585	1.887	1.500
	PSNR (dB)	21.13	21.24	24.63	22.02	29.24	35.27	41.26	45.43	35.20	41.26	47.19	47.70
	SSIM	0.6549	0.6510	0.7904	0.8399	0.7825	0.9093	0.9713	0.9890	0.9088	0.9707	0.9922	0.9955
Barbara	Total bits	494284	473225	333301	590112	623713	437551	307388	318270	820322	628937	468163	393217
	bpp	1.886	1.806	1.271	2.251	2.379	1.669	1.173	1.214	3.129	2.399	1.786	1.500
	PSNR (dB)	23.30	23.59	26.79	24.10	30.19	36.23	42.02	45.44	36.18	42.22	47.27	46.47
	SSIM	0.7562	0.7570	0.8435	0.8614	0.6582	0.8025	0.9012	0.9425	0.8047	0.9087	0.9600	0.9658
Boats	Total bits	455617	444284	303200	440769	687627	509241	356615	260458	881426	695334	525975	393217
	bpp	1.738	1.695	1.157	1.681	2.623	1.943	1.360	0.994	3.362	2.652	2.006	1.500
	PSNR (dB)	26.60	26.59	30.88	29.15	28.68	34.59	40.46	45.43	34.53	40.52	45.96	48.05
	SSIM	0.7840	0.7870	0.8759	0.9493	0.5738	0.7674	0.8976	0.9604	0.7696	0.9009	0.9615	0.9793
Bridge	Total bits	590842	568452	392396	733636	618802	435031	301438	323632	813294	624163	462801	393217
	bpp	2.254	2.169	1.497	2.799	2.361	1.660	1.150	1.235	3.102	2.381	1.765	1.500
	PSNR (dB)	23.57	23.69	27.57	22.23	30.31	36.40	42.14	45.39	36.35	42.39	47.42	46.43
	SSIM	0.7204	0.7180	0.8435	0.8479	0.8362	0.9345	0.9755	0.9873	0.9361	0.9787	0.9924	0.9938

Table 4.4: Performance Comparison of the Proposed Scheme (Contd.)

Test Images	Performance Metric	Jung & Yoo [45]	Lee & Huang [46]	Chang *et al.* [48]	Zhang *et al.* [49]	embedding with Flag (Proposed)				embedding without Flag (Proposed)			
						T=3	T=4	T=5	T=6	T=3	T=4	T=5	T=6
Couple	Total bits	423485	407754	284262	586075	667161	473734	305315	299875	861865	667800	486558	393217
	bpp	1.615	1.556	1.084	2.236	2.545	1.807	1.165	1.144	3.288	2.547	1.856	1.500
	PSNR (dB)	25.59	25.48	28.50	27.43	29.58	35.59	41.52	45.44	35.51	41.53	47.45	47.56
	SSIM	0.8224	0.8200	0.8987	0.9230	0.5986	0.7858	0.9081	0.9536	0.7838	0.9069	0.9690	0.9788
Elaine	Total bits	432844	415308	302213	546098	688285	491972	318981	282726	884891	688432	503707	393217
	bpp	1.651	1.585	1.153	2.083	2.626	1.877	1.217	1.079	3.376	2.626	1.921	1.500
	PSNR (dB)	29.78	29.86	33.29	30.78	28.92	35.04	41.02	45.42	34.97	41.04	47.03	48.00
	SSIM	0.7327	0.7470	0.8295	0.9255	0.5516	0.7640	0.9040	0.9628	0.7621	0.9032	0.9679	0.9864
Goldhill	Total bits	447244	438737	293630	564789	613016	422962	288759	333906	809625	616294	452527	393217
	bpp	1.706	1.674	1.120	2.155	2.339	1.614	1.102	1.274	3.088	2.351	1.726	1.500
	PSNR (dB)	28.32	28.38	32.25	29.20	30.63	36.84	42.54	45.42	36.79	42.80	47.87	46.25
	SSIM	0.7974	0.7970	0.8843	0.9285	0.6995	0.8454	0.9229	0.9573	0.8458	0.9269	0.9664	0.9795
Lena	Total bits	396268	380774	248982	470653	656093	460611	326825	294693	852696	656654	491740	393217
	bpp	1.512	1.453	0.950	1.795	2.503	1.757	1.247	1.124	3.253	2.505	1.876	1.500
	PSNR (dB)	29.65	29.61	34.86	31.52	29.43	35.38	41.36	45.44	35.31	41.52	46.88	47.13
	SSIM	0.8660	0.8670	0.9244	0.9546	0.5064	0.6818	0.8314	0.9207	0.6799	0.8327	0.9352	0.9678



(a) (b) (c)

(d) (e) (f)

Figure 4.3: Example of embedded images of: (a) Jung & Yoo, (b) Lee & Huang, (c) Chang *et al*., (d) Zhang *et al*., (e) proposed embedding with flag and & $T = 3$, and (f) proposed embedding without flag and & $T = 3$,

ity requirement and have demonstrated their highest embedding capacity with respective embedded image quality. So, for the maximum embedding capacity of the other schemes, we compare their embedded image quality with that of the proposed scheme. A few relevant observations are thus noted below.

(a) As Table 4.5 presents, among all the considered schemes, Zhang *et al*.'s [49] scheme has demonstrated the maximum average embedding capacity of 585 Kbit (or 2.234 bpp) with PSNR of 26.55 dB, where the proposed scheme with flag have the embedding capacity more than 660 Kbit (or 2.541 bpp) with a higher PSNR of 29.40 dB for $T = 3$. This trend of higher embedding rate and lower distortion of our scheme with $T = 3$ are also evident for individual test images detailed in Table 4.3 and Table 4.4.

(b) In other words, while the embedding rate of the proposed scheme with flag increases 36.4%, 40.0%, 104.1% and 14.1% in average compared to the other schemes [45, 46, 48, 49], PSNR values are also evident to be 14.1%, 13.9%, -1.0% (1% less) and 13.7% higher, respectively for $T = 3$. With this significant improvement in embedding capacity and PSNR of our scheme at $T = 3$, the SSIM values remain lower than the other schemes. This is because, the MSE of our scheme usually remains lower than the other schemes showing relative improvement in PSNR, while the co-variance of the proposed scheme is higher than the other schemes resulting in relatively lower SSIM values. It is to include that, Fig. 4.3 illustrates that having higher PSNR and lower SSIM values with $T = 3$, the embedded images of the proposed scheme (both with and without flag) remains visually almost similar compared to the other schemes (Although this comparison is shown only for Elaine image, the similar visual quality is also evident for other test images). However, significant improvement of SSIM and PSNR based image quality compared to other schemes, can be effectively managed by our scheme with higher values of T (*e.g.*, $T = 4$), if required with relatively lower embedding capacity.

(c) The performance variation with T can further be analyzed from Table 4.5. For example, with $T = 4$, while the average embedding rate of the scheme with flag gets reduced by 2.5% and 41.9% than the schemes in [45] and [49], The rate is still 0.1% and 45% higher than that in [46] and [48], respectively. At the same time, with $T = 4$, the PSNR value increases 37.5%, 37.3%, 19.2% and 26.55% compared to the schemes in [45], [46], [48] and [49], respectively.

(d) The rate-distortion performance of our proposed scheme can be efficiently controlled with the value of T because the proposed scheme can embed within a broader dynamic range, which is not the case in any of the considered schemes.

Table 4.5: Comparison of Average Rate-Distortion Performance

Scheme		**Capacity/ rate**		**Visual Quality**	
		Total bits	**bpp**	**PSNR (dB)**	**SSIM**
Jung & Yoo [45]		488265	1.863	25.755	0.7709
Lee & Huang [46]		475313	1.814	25.80	0.7712
Chang *et al*. [48]		326453	1.245	29.71	0.8661
Zhang *et al*. [49]		585595	2.234	26.55	0.8981
(Proposed) Embedding with Flag	T=3	666055	2.541	29.40	0.6507
	T=4	475719	1.815	35.43	0.8115
	T=5	321748	1.227	41.32	0.9140
	T=6	291160	1.111	45.43	0.9605
(Proposed) Embedding without Flag	T=3	861874	3.288	35.37	0.8114
	T=4	668797	2.551	41.42	0.9158
	T=5	495273	1.889	46.97	0.9682
	T=6	393217	1.500	47.61	0.9821

For example, it can effectively embed from 291 Kbit to 666 Kbit of data in an interpolated image of size $512 \times 512 \times 8$, with a PSNR value ranging from 29 dB to 45 dB, and a SSIM value ranging from 0.65 to 0.96 for $T \in \{3, 4, 5, 6\}$. Whereas, Jung and Yoo [45] embeds 488 Kbit with PSNR of 25.755 dB and SSIM of 0.7709; Lee and Huang [46] embeds 475 Kbit with PSNR of 25.8 dB and SSIM of 0.7712; Chang *et al*. [48] embeds 326 Kbit with PSNR of 29.7 dB and SSIM of 0.866 and Zhang *et al*. [49] embeds 585 Kbit with PSNR of 26.55 dB and SSIM of 0.8981.

The above observations suggest that the proposed scheme with flag would have higher embedding capacity (or rate) with better embedded image quality than the other schemes for natural images, with a suitable value of T.

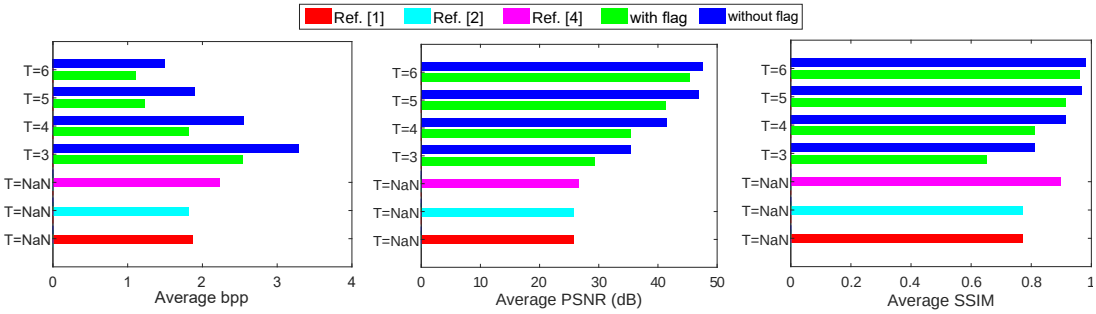


Figure 4.4: Average performance comparison of the proposed scheme with other schemes for different values of T in terms of: (a) bpp, (b) PSNR, and (c) SSIM.

4.4.2 Performance of the proposed scheme embedded without flag

The proposed scheme embedded without flag also outperforms the prominent IRDH schemes in [45, 46, 48, 49] as evident for the individual test images in Table 4.3 and 4.4. This improvements become more apparent in the average performance given in Table 4.5. Some important observations supporting the improvement by the proposed scheme without flag are noted below.

(a) The range of embedding capacity increases with the proposed scheme without flag, which helps better control over the rate-distortion performance with different values of T, which is also not the case in any of the considered schemes in [45,46,48,49]. As mentioned earlier in the previous subsection and evident from Table 4.5, a maximum of 488 Kbit with PSNR of 25.755 dB and SSIM of 0.7709 is embedded by Jung and Yoo [45]; 475 Kbit with PSNR of 25.8 dB and SSIM of 0.7712 is embedded by Lee and Huang [46]; 326 Kbit with PSNR of 29.71 dB and SSIM of 0.866 by is embedded Chang *et al*. [48] and 585 Kbit with PSNR of 26.55 dB and SSIM of 0.8981 is embedded by Zhang *et al*. [49]. Whereas the proposed scheme without flag can effectively embed from 393 Kbit to 861 Kbit of data in an interpolated image of size $512 \times 512 \times 8$, with a PSNR value ranging from 35 dB to 47 dB, and a SSIM value ranging from 0.81 to 0.98 for $T \in \{3, 4, 5, 6\}$.

(b) Table 4.5 demonstrates that the Zhang *et al.*'s scheme [49] offers the highest embedding rate-among the schemes in [45, 46, 48, 49], which can embed an average size payload of 585 Kb (or embedding rate of 2.234 bpp) with PSNR of 26.55 dB and SSIM of 0.898. For our proposed scheme, considering its closer embedding rate to that of the Zhang *et al.*'s scheme, we see that an embedding rate of 2.551 bpp (total 668 Kbit) is obtained with a better PSNR of 41.42 dB and SSIM of 0.9158 for $T = 4$. Thus with $T = 4$, our proposed scheme offers 14% higher embedding rate with 56% and 2% better PSNR and SSIM, respectively. Moreover, we also note that the requirement of either a higher capacity or a better image quality can be attained with a lower or higher value of T, respectively for our proposed IRDH scheme.

(c) The rate-distortion performance of our scheme is analyzed with a varying T to demonstrate its efficiency to manage higher embedding rate with reasonable PSNR and SSIM values. For example, with $T = 5$, our scheme offers better embedded image quality for a lower embedding rate. It is evident that the embedding rate is lower in this case than the Zhang *et al.*'s [49] scheme. The embedding rate however still remains higher than the Jung and Yoo's [45], Lee and Huang's [46] and Chang *et al.*'s [48] scheme with better image quality (*i.e.*, higher PSNR and SSIM) compared to all the schemes for almost all the test images. Similarly, with $T = 3$, our scheme offers significantly higher embedding rate than other schemes with a similar (for Baboon, Barbara, Bridge and Couple images) or lower (for other images) PSNR values.

4.4.3 Comparison between embedding with and without flag

The adaptive embedding technique in Sec. 3.4.2 is modelled and proposed to avoid the use of flag bit unlike the proposed technique in Sec. 3.4.1, so that one more payload bit can be embedded replacing a flag-bit in an embeddable pixel. To compare the performance between the two proposed embedding techniques (*i.e.*, with and without flag), some observations are pointed below.

(a) The improvement of embedding rate and capacity with better or almost similar embedded image quality becomes evident with the proposed embedding technique without flag, while it is compared to the proposed embedding with flag for the same values of T as in Fig. 4.4 and Fig. 4.5.

(b) As demonstrated in Table 4.4, with $T = 4$, the proposed embedding with flag can embed up to 460611 bit payload with 1.76 bpp in Lena image, while the PSNR is 35.38 dB and SSIM is 0.6818; whereas, 656654 bit payload (2.51 bpp) with better image quality (*i.e.*, PSNR of 41.52 dB and SSIM of 0.8327) are embedded using the proposed embedding without flag for the same value of T. This means that for a given value of T, embedding without flag achieves higher embedding rate with improved image quality compared to the embedding with flag.

(c) For higher embedding capacity requirement (*i.e.*, lower values of T), embedding without flag also demonstrates better rate-distortion performance than embedding with flag. For instance, with $T = 5$ and Lena image, embedding without flag embeds total 491740 bits with 1.88 bpp, PSNR of 46.88 dB and SSIM of 0.9352. In contrast, with $T = 4$ and Lena image, embedding with flag can embed total 460611 bits of payload with 1.76 bpp, PSNR of 35.38 dB and SSIM of 0.6818, which are lower than the proposed embedding without flag. This trend of improvement is also evident for the other test images as in Fig. 4.5.

(d) Statistically, the scheme embedded with flag achieves an average maximum capacity ranging from 291 Kb to 666 Kb, PSNR from 29.40 dB to 45.43 dB and SSIM from 0.65 to 0.96 for $T = [3, 6]$. In contrast, the average embedding ca-

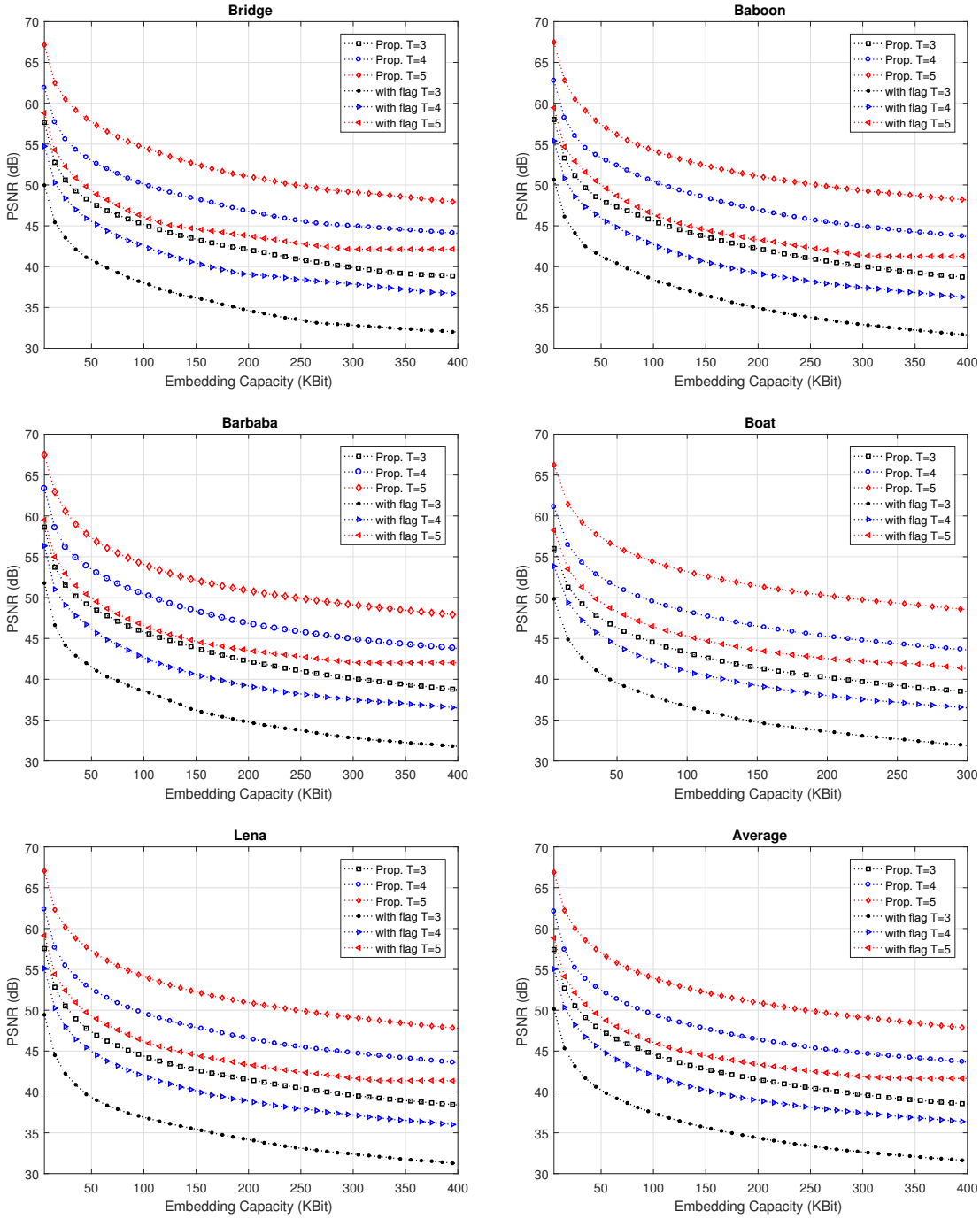


Figure 4.5: Embedding rate-distortion performance comparison of the proposed (without flag and with flag) schemes for different values of T: (a) Bridge, (b) Baboon, (c) Barbara, (d) Boat, (e) Lena and (f) average of all test-images.

pacity of the scheme embedded without flag varies in the dynamic range from 393 Kb to 861 Kb payload (*i.e.*, 1.50 bpp to 3.29 bpp) for the same range of the capacity control parameter, with PSNR and SSIM values ranging from 35.37 dB to 47.61 dB and 0.811 to 0.982, respectively. With $T = 5$, for example, this improvement is recorded with 53.9% higher embedding capacity, 13.67% higher PSNR, and 5.93% higher SSIM, which are illustrated for the all considered values of T in Fig. 4.4.

In summary, the proposed scheme with computationally efficient SPI and adaptive embedding (both with and without flag) is better than the other existing prominent IRDH schemes [45, 46, 48, 49], (see Fig. 4.4).

However, the above observations suggest that the proposed scheme without flag offers higher dynamic range of embedding capacity and better embedded image quality compared to the proposed scheme with flag. Considering the rate-distortion performance, the proposed IRDH scheme without flag outperforms the proposed scheme with flag. Compared to the scheme with flag, the embedding capacity improvements are recorded 29%, 40%, 53% and 35% higher with 20%, 16.9%, 13.6% and 4.8% higher PSNR and 24.7%, 12.9%, 5.9% and 2.24% higher SSIM for $T = 3, 4, 5 \ \& \ 6$, respectively.

4.5 Chapter Summary

The performance of the proposed IRDH scheme is evaluated and validated for USC-SIPI test images. The embedding capacity and rate are evaluated in terms of total bits embedded and bpp, respectively. The image quality is expressed in terms of PSNR and SSIM. The SPI technique for image up-sampling is demonstrated to be computationally efficient. The new adaptive embedding technique demonstrates its effectiveness to

offer better rate-distortion performance compared to other prominent IRDH schemes. In other words, the proposed scheme has demonstrated that it can effectively embed varying size payloads with higher embedding rate and better image quality with a suitable capacity control parameter. Based on the observations presented in this chapter, concluding remarks and future works of the research are presented in the following chapter.

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

The research presented in this thesis aimed to develop an adaptive IRDH scheme for digital image applications. IRDH schemes have created a new paradigm in RDH research. The performance of an IRDH scheme is determined by the effectiveness of its two main processes: (i) interpolation (ii) payload embedding. Interpolation is used in IRDH scheme to up-sample a cover image to a higher spatial resolution. The interpolated image, specifically the interpolated pixels are then used for embedding payload. All the existing interpolation techniques do not offer similar interpolated image quality and are not equally computationally efficient. So, a better interpolation technique is an important requirement for embedding. For embedding payload, different approaches are reported such as DLE, PEE and their combination. Among different embedding techniques, DLE offers greater simplicity, better user access control and better embedding rate-distortion performance over its counterparts. Thus the scope of the research presented in this thesis is focused on the development of DLE based IRDH schemes.

Despite the great promises of the existing IRDH schemes with DLE, they do not account for embedding varying size payloads required in many practical application scenarios. Thus they compromise the maximum possible embedded image quality due to non-uniform distribution of distortion in the embedded image. For embedding of varying size payloads, they might result in the same level of distortion. That means, they may fail to maintain the best possible rate-distortion performance for different

size of payloads, which is not desirable. This varying capacity requirement can be effectively managed by using a suitably designed adaptive data embedding technique.

Therefore, a number of original contributions have been made. A new IRDH scheme is developed and validated with computationally efficient up-sampling and adaptive embedding to tackle the above mentioned varying capacity requirement attainment problem of the existing IRDH schemes. Particularly, a computationally efficient interpolation technique (SPI technique) has been developed with reasonably better image quality and is utilized for computing interpolated pixels. A capacity control parameter for varying capacity requirements is formulated with the consideration of embedding capacity requirement.

Additionally, an effective embedding technique has been developed, which mainly embeds the payload bits in the LSBs of the interpolated pixels either in original or their complement form. This idea helps minimize the embedding distortion. Different values of the capacity control parameter is considered for attaining a dynamic range of embedding rate to fulfill the varying capacity requirement. A flag bit is used to track whether the form of embedded bits (original or complement). The proposed embedding with flag demonstrates better embedding-rate distortion performance compared to other IRDH schemes. Particularly, it can effectively embed payload of size 291 Kb to 666 Kb in a cover image of size $256 \times 256 \times 8$ with a reasonably higher PSNR and SSIM compared to other IRDH schemes.

Besides, the embedding capacity is further significantly increased avoiding the use of the flag bit during embedding. The logical correlation between the embeddable pixel and estimated versions of the embedded pixel are reinvestigated and a new adaptive embedding technique without flag is developed. It adaptively selects the value of the capacity control parameter for attaining varying embedding capacity requirement with a minimum possible distortion in the embedded image and without using any flag bit. Experimental results have substantiated that the proposed embedding without flag can

effectively embed varying size payload with significantly higher embedding rate and better image quality compared to the proposed embedding with flag. While both the two proposed embedding techniques are evident to perform better than the existing embedding techniques in IRDH schemes, between those two, the proposed embedding without flag outperforms the proposed scheme with flag. For embedding without flag, the embedding capacity improvements are recorded 29%, 40%, 53% and 35% higher with 20%, 16.9%, 13.6% and 4.8% higher PSNR and 24.7%, 12.9%, 5.9% and 2.24% higher SSIM for different values of the control parameter.

The presented IRDH scheme has a number of advantages over the existing schemes. For example, it can offer a higher embedding capacity with better image quality, *i.e.*, better rate-distortion performance compared to other recent prominent IRDH schemes. Embedding is carried out only in the interpolated pixels that would also minimize any possible concern of erratically modifying the input image and thus the proposed scheme could be useful in military and medical image applications that restrict minimum possible changes in a cover image. Extraction is modelled to blindly extract the embedded payload and to completely recover the original image, where these two processes were kept mutually independent to ensure better user control in an application scenario. This also means that the cover image can be instantly recovered with or without extracting the embedded data. Additionally, being up-sampled, the embedded image would have a higher spatial resolution. It also does not require any location map, and thus the total capacity can be effectively used for data embedding.

The proposed adaptive IRDH scheme and its evaluation demonstrate better rate-distortion performance for attaining the requirement of varying capacity management effectively, in regards to which the following future works are addressed.

5.2 Future Works

Some possible avenues for future research have been identified. These are as follows. Computational complexity analysis of the proposed adaptive embedding may be carried out in future towards developing a more computationally efficient adaptive embedding technique for IRDH scheme.

Evaluation of the performance of the proposed scheme in an extended experimental set up and comparison with the other prominent IRDH schemes can be shown in future. The proposed IRDH scheme can be implemented and validated for a practical application scenario such as medical image applications, electronic patient record management, *etc*. The proposed scheme can be applied and validated for color image applications. Further improvement of the embedded image quality of the presented IRDH scheme can be reinvestigated minimizing the local dispersion in the embedded images.

LIST OF PUBLICATIONS

Journal Papers:

(i) **M. A. Wahed** and H. Nyeem, "Reversible data hiding with interpolation and adaptive embedding," *Multimedia Tools and Applications*, Springer, 2018.

(ii) **M. A. Wahed** and H. Nyeem, "High capacity reversible data hiding with interpolation and adaptive embedding," *PLOS One*, PLOS, 2017, (minor revision).

Conference Papers:

(iii) **M. A. Wahed** and H. Nyeem, "Developing a block-wise interpolation based adaptive data embedding scheme," in *Proc. of ICEEICT 2016*, IEEE, 2016, pp. 1–6 (**best paper award**).

(iv) **M. A. Wahed** and H. Nyeem, "A simplified parabolic interpolation based reversible data hiding scheme," in *Proc. of ICAEE 2017*, IEEE, 2017.

(v) **M. A. Wahed** and H. Nyeem, "Modeling and analysis of interpolation based adaptive reversible data hiding," in *Proc. of EICT 2017*, IEEE, 2017.

(vi) **M. A. Wahed** and H. Nyeem, "Efficient data embedding for interpolation based reversible data hiding scheme," in *Proc. of ICEEE 2017*, IEEE, 2017.

(vii) **M. A. Wahed** and H. Nyeem, "Efficient LSB substitution for interpolation based reversible data hiding scheme," in *Proc. of ICCIT 2017*, IEEE, 2017 (**best paper award**).

BIBLIOGRAPHY

[1] J. Welbes, "Edina medical imaging company sued for insurance fraud," https://www.twincities.com/2013/10/14/edina-medical-imaging-company-sued-for-insurance-fraud/, [Online; posted on 7-Nov-2015].

[2] Matt Wright, "Doctor sentenced for insurance fraud scheme," http://www.abc-7.com/story/23140026/doctor-sentenced-for-insurance-fraud-scheme/, [Online; posted on 15-Aug-2013].

[3] "Allstate wins multi-million dollar insurance fraud lawsuit," http://www.wmcactionnews5.com/story/19881952/doctor-arrested-on-charges/, [Online; posted on 26-Sep-2013].

[4] R. Smith, "Nj doctor headed to prison for 3 million insurance fraud," https://www.insurancebusinessmag.com/us/news/healthcare/nj-doctor-headed-to-prison-for-3-million-insurance-fraud-96176.aspx/, [Online; posted on 28-Mar-2018].

[5] Albany, "Doctor sentenced to prison for participating in 30m scheme to defraud medicare and medicaid," https://www.insurancefraud.org/IFNS-detail.htm?key=28906/, [Online; posted on 01-Aug-2018].

[6] Jamaica News, "St elizabeth doctor sentenced to 12 months in prison for fraud," http://www.loopjamaica.com/content/st-elizabeth-doctor-sentenced-12-months-prison-fraud/, [Online; posted on 17-Feb-2018].

[7] W. Bender, W. Butera, D. Gruhl, R. Hwang, F. J. Paiz, and S. Pogreb, "Applications for data hiding," *IBM systems journal*, vol. 39, no. 3.4, pp. 547–568, 2000.

[8] A. H. Tewfik and M. Swanson, "Data hiding for multimedia personalization, interaction, and protection," *IEEE Signal Processing Magazine*, vol. 14, no. 4, pp. 41–44, 1997.

[9] G. B. Rhoads, "Signal processing to hide plural-bit information in image, video, and audio data," Sep. 19 2000, uS Patent 6,122,392.

[10] M. Arnold, M. Schmucker, and S. D. Wolthusen, *Techniques and applications of digital watermarking and content protection*. Boston :: Artech House, 2003.

[11] DWA, "Digital watermarking alliance (DWA)," http://digitalwatermarkingalliance.org/digital-watermarking-applications/, 2006, [Online; last accessed: Jan 2017].

[12] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, "Models of watermarking," in *Digital Watermarking and Steganography (Second Edition)*. Burlington: Morgan Kaufmann, 2008, pp. 61–103.

[13] H. Nyeem, W. Boles, and C. Boyd, "Digital image watermarking: its formal model, fundamental properties and possible attacks," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 1–22, 2014.

[14] J. M. Barton, "Method and apparatus for embedding authentication information within digital data," Jul. 8 1997, US Patent 5,646,997.

[15] C. Rey and J.-L. Dugelay, "A survey of watermarking algorithms for image authentication," *Eurasip Journal on Applied Signal Processing*, vol. 2002, no. 1, pp. 613–621, 2002.

[16] G. Voyatzis and I. Pitas, "The use of watermarks in the protection of digital multimedia products," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1197–1207, 1999.

[17] S. A. Parah, F. Ahad, J. A. Sheikh, and G. M. Bhat, "Hiding clinical information in medical images: a new high capacity and reversible data hiding technique," *Journal of biomedical informatics*, vol. 66, pp. 214–230, 2017.

[18] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1167–1180, 1999.

[19] Y. Lee, H. Kim, and Y. Park, "A new data hiding scheme for binary image authentication with small image distortion," *Information Sciences*, vol. 179, no. 22, pp. 3866–3884, 2009.

[20] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation," *IEEE transactions on multimedia*, vol. 6, no. 4, pp. 528–538, 2004.

[21] H. H. Yu, M. Wu, X. Li, and A. D. Gelman, "Methods and apparatus for multi-layer data hiding," Sep. 24 2002, uS Patent 6,456,726.

[22] A. K. Singh, "Improved hybrid algorithm for robust and imperceptible multiple watermarking using digital images," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8881–8900, 2017.

[23] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding-new paradigm in digital watermarking," *EURASIP Journal on Applied Signal Processing*, pp. 185–196, 2002.

[24] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-lsb data embedding," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253–266, 2005.

[25] K.-H. Jung and K.-Y. Yoo, "Steganographic method based on interpolation and lsb substitution of digital images," *Multimedia Tools and Applications*, vol. 74, no. 6, pp. 2143–2155, 2015.

[26] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 890–896, 2003.

[27] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, pp. 1147–1156, 2004.

[28] Y.-C. Liu, H.-C. Wu, and S.-S. Yu, "Adaptive de-based reversible steganographic technique using bilinear interpolation and simplified location map," *Multimedia Tools and Applications*, vol. 52, no. 2-3, pp. 263–276, 2011.

[29] C.-W. Shiu, Y.-C. Chen, and W. Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion," *Signal Processing: Image Communication*, vol. 39, pp. 226–233, 2015.

[30] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, pp. 989–999, 2009.

[31] T.-C. Lu, C.-C. Chang, and Y.-H. Huang, "High capacity reversible hiding scheme based on interpolation, difference expansion, and histogram shifting," *Multimedia tools and applications*, vol. 72, no. 1, pp. 417–435, 2014.

[32] C.-T. Wang and H.-F. Yu, "High-capacity reversible data hiding based on multi-histogram modification," *Multimedia Tools and Applications*, vol. 61, no. 2, pp. 299–319, 2012.

[33] Z. Pan, S. Hu, X. Ma, and L. Wang, "Reversible data hiding based on local histogram shifting with multilayer embedding," *Journal of Visual Communication and Image Representation*, vol. 31, pp. 64–74, 2015.

[34] J. Wang, J. Ni, X. Zhang, and Y.-Q. Shi, "Rate and distortion optimization for reversible data hiding using multiple histogram shifting." *IEEE Trans. Cybernetics*, vol. 47, no. 2, pp. 315–326, 2017.

[35] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 255–258, 2007.

[36] X. Chen, X. Li, B. Yang, and Y. Tang, "Reversible image watermarking based on a generalized integer transform," in *Proc. ICASSP*. IEEE, 2010, pp. 2382–2385.

[37] K. Aiswaria, G. George, M. Murali, K. Nimitha, B. Sangeetha, and B. Rapheal, "Reversible image data hiding with contrast enhancement," *International Journal of Engineering and Management Research (IJEMR)*, vol. 7, no. 3, pp. 164–171, 2017.

[38] H.-T. Wu, S. Tang, J. Huang, and Y.-Q. Shi, "A novel reversible data hiding method with image contrast enhancement," *Signal Processing: Image Communication*, vol. 62, pp. 64–73, 2018.

[39] D. M. Thodi and J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, pp. 721–730, 2007.

[40] M. Fallahpour, D. Megias, and M. Ghanbari, "Subjectively adapted high capacity lossless image data hiding based on prediction errors," *Multimedia Tools and Applications*, vol. 52, no. 2-3, pp. 513–527, 2011.

[41] X. Ma, Z. Pan, S. Hu, and L. Wang, "High-fidelity reversible data hiding scheme based on multi-predictor sorting and selecting mechanism," *Journal of Visual Communication and Image Representation*, vol. 28, pp. 71–82, 2015.

[42] X. Chen, X. Sun, H. Sun, L. Xiang, and B. Yang, "Histogram shifting based reversible data hiding method using directed-prediction scheme," *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5747–5765, 2015.

[43] B. Ou, X. Li, and J. Wang, "High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion," *Journal of Visual Communication and Image Representation*, vol. 39, pp. 12–23, 2016.

[44] S. Yi, Y. Zhou, and Z. Hua, "Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion," *Signal Processing: Image Communication*, vol. 64, pp. 78–88, 2018.

[45] K.-H. Jung and K.-Y. Yoo, "Data hiding method using image interpolation," *Computer Standards & Interfaces*, vol. 31, no. 2, pp. 465–470, 2009.

[46] C.-F. Lee and Y.-L. Huang, "An efficient image interpolation increasing payload in reversible data hiding," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6712–6719, 2012.

[47] W.-J. Wang, Y.-H. Zhang, C.-T. Huang, and S.-J. Wang, "Steganography of data embedding in multimedia images using interpolation and histogram shifting," in *Proc of IIHMSP'13*. IEEE, 2013, pp. 387–390.

[48] Y.-T. Chang, C.-T. Huang, C.-F. Lee, and S.-J. Wang, "Image interpolating based data hiding in conjunction with pixel-shifting of histogram," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1093–1110, 2013.

[49] X. Zhang, Z. Sun, Z. Tang, C. Yu, and X. Wang, "High capacity data hiding based on interpolated image," *Multimedia Tools and Applications*, pp. 1–24, 2016.

[50] A. Malik, G. Sikka, and H. K. Verma, "An image interpolation based reversible data hiding scheme using pixel value adjusting feature," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 1–22, 2016.

[51] ——, "Image interpolation based high capacity reversible data hiding scheme," *Multimedia Tools and Applications*, vol. 76, no. 22, pp. 7–23, 2017.

[52] J. Hu and T. Li, "Reversible steganography using extended image interpolation technique," *Computers & Electrical Engineering*, vol. 46, pp. 447–455, 2015.

[53] X.-T. Wang, C.-C. Chang, T.-S. Nguyen, and M.-C. Li, "Reversible data hiding for high quality images exploiting interpolation and direction order mechanism," *Digital Signal Processing*, vol. 23, no. 2, pp. 569–577, 2013.

[54] D. Han, "Comparison of commonly used image interpolation methods," *Proc. of ICCSEE'13*, pp. 1556–1559, 2013.

[55] H. Nyeem, W. Boles, and C. Boyd, "A review of medical image watermarking requirements for teleradiology," *J. of Dig. Imaging*, vol. 26, pp. 326–343, 2013.

[56] S. A. Parah, F. Ahad, J. A. Sheikh, N. A. Loan, and G. M. Bhat, "A new reversible and high capacity data hiding technique for e-healthcare applications," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 3943–3975, 2017.

[57] G. Coatrieux, L. Lecornu, B. Sankur, and C. Roux, "A review of image watermarking applications in healthcare," in *Proc. of IEEE EMBS'06*. IEEE, 2006, pp. 4691–4694.

[58] H. Nyeem, W. Boles, and C. Boyd, "Utilizing least significant bit-planes of roni pixels for medical image watermarking," in *Proc. DICTA*. IEEE, 2013, accepted.

[59] ——, "Content-independent embedding scheme for multi-modal medical image watermarking," *Biomedical engineering online*, vol. 14, no. 1, p. 7, 2015.

[60] ——, "Developing a digital image watermarking model," in *Proc. DICTA*. IEEE, 2011, pp. 468–473.

[61] USC, "SIPI image database," http://sipi.usc.edu/database/, SIPI, [Online; last accessed 23-Mar-2017].

[62] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.