JJEE
Jordan Journal of Electrical Engineering

Volume 7, Number 2, June 2021
Pages 130-146
ISSN (Print): 2409-9600, ISSN (Online): 2409-9619

A Low Distortion Reversible Data Hiding Technique Based on Prediction Difference Expansion

Ahmad A. Mohammad*

Electrical Engineering Department, School of Engineering, Princess Sumaya University for Technology,
Amman, Jordan
E-mail: atawayha@psut.edu.jo

Received: October 02, 2020 *Revised: December 30, 2020* *Accepted: January 8, 2021*

Abstract— Difference expansion based data hiding techniques provide relatively high data hiding capacity. However, increasing the data hiding capacity introduces more distortion to the cover image. A solution to this problem is the use of a distortion control parameter (*T*) that limits the distortion to cover image. Unfortunately, this will decrease the data hiding capacity. Another factor that affects both data hiding capacity and distortion is the predictor utilized to predict pixel values. This paper presents a new prediction technique that reduces distortion to cover image. In contrast to common predictors of difference expansion techniques that utilize one or two adjacent pixels for prediction, the proposed technique utilizes three or four adjacent pixels. This technique uses about one-fourth of cover image pixels as reference pixels without any distortion and thus, reducing the cover image distortion and resulting in high-quality stego images. The proposed technique achieved average peak signal to noise ratio (PSNR) values in the range 33.29-49.7 dB. The average embedding rate in bits per pixel (bpp) ranged from 0.09 to 0.59 bpp. Moreover, the proposed technique showed to be reversible and does not require a location map.

Keywords— Reversible data hiding; Interpolation; Low distortion; Difference expansion; Hiding capacity; embedding rate.

1. INTRODUCTION

In the last two decades, the size of digital content has been vastly increasing. In addition, tremendous amounts of digital content are transmitted over the internet. The easiness of tampering, sharing and replication of digital content necessitates the use of techniques for ownership protection and security of secret information. In some instances, there is a need to, secretly, share digital information. Data hiding techniques, transparently, insert a secret message in the digital content (cover object) to provide ownership protection or to hide information. This is referred to as digital watermarking when the objective is copyright protection and steganography when the objective is secret communication. Insertion of the secret message into the cover object results in a distorted stego-object with reduced visual quality. It is important to keep the distortion minimal and transparent so that intruders cannot detect it. In addition to transparency, another important factor in steganography is the amount of secret information that one can insert in the cover media. This is termed as the data-hiding capacity. Unfortunately, increasing data hiding capacity increases distortion and reduces visual quality. Thus, one needs to make a compromise between them. When the distortion incurred to the cover object is not permanent and one can restore the cover object after extracting the secret message, the data hiding technique (DHT) is classified as a reversible data hiding technique (RDHT). If the distortion is permanent and the cover object cannot be restored after extraction of the secret data, the DHT is classified as irreversible DHT. RDHTs find many application areas such as military, law enforcement and

* Corresponding author

Articles' DOI: 10.5455/jjee.204-1601584525

medical applications where permanent distortion is not acceptable. The last two decades witnessed extensive research in this area [1-39]. Early RDHTs [23-26] utilized spread spectrum addition to embed the secret data in the cover object and suffered from salt and pepper artifact. They also have limited data hiding capacity. Solutions to salt and pepper artifact were proposed in [27-29]. A remarkable increase, in data hiding capacity in RDHTs by the introduction of difference expansion algorithms, was proposed in [22, 30-34]. These algorithms expand a feature in the cover object to make space to insert the secret data. Another type of RDHTs relies on histogram modification to insert the secret data [35-39]. This type of RDHTs is known to have a limited data hiding capacity because it needs to embed a location map in conjunction of the secret data. Tian [22] introduced a difference expansion based RDHT that can provide a data hiding capacity of 0.5 bits per pixel (bpp). Unfortunately, it suffered over from underflow problems that required a location map to be embedded together with the secret message reducing the actual embedding capacity. In [15], Tseng et al. developed a difference expansion prediction-based RDHT that does not require a location map for cover image recovery and thus, increasing pure data hiding capacity. In [4], Chen et al. modified Tseng's algorithm to increase the embedding capacity by increasing the number of embeddable pixels. Both Tseng and Chen Feng methods utilized a control parameter T to control the level of distortion and embedding capacity. They also used two adjacent pixels for predicting the value of a pixel. In both algorithms, some pixels of the cover image are used as reference pixels that do not change in the embedding process. The increase in the number of reference pixels reduces data hiding capacity and increases visual quality. The accuracy of predicted value has a direct effect on the data hiding capacity and the level of distortion introduced to the cover image. Thus, the more accurate the prediction is the less distortion and higher embedding capacity.

In this paper, we propose a prediction difference expansion RDHT with a new prediction algorithm. Unlike Tseng and Chen that use two adjacent points, our prediction algorithm uses three and four adjacent pixels to find a prediction value of a pixel. Our technique uses a larger number of reference pixels resulting in a vast improvement in the stego image quality while slightly decreasing data hiding capacity. In situations where visual quality is a high priority, the proposed technique gives an adjustable means to provide high-quality stego images with moderate data hiding capacity.

The rest of this paper proceeds as follows: section 2 gives a brief summary of related work. Section 3 outlines the prediction, the data-hiding and the data extraction with image recovery step. Section 4 gives experimental simulation results and section 5 gives the conclusions.

2. RELATED WORK

In this section, we present two closely related RDHTs: the first one was proposed by Tseng and Hsieh [15] and the second was proposed by Chen et al. [4].

2.1. Tseng and Hsieh Scheme

Given a cover image I , let $I(x,y)$ denote the gray level of the pixel with coordinates (x,y). Also, let $\hat{I}(x,y)$ denote the prediction value of that pixel. The embedding algorithm starts by finding the prediction values for all pixels except for the first row and first column of I that remain intact and referred to as wall pixels or as reference pixels. The prediction process proceeds in a raster scan from left to right and top to bottom. It calculates the prediction value as: $\hat{I}(x,y)=\lfloor (I(x-1,y)+I(x,y-1))/2 \rfloor$. Next, it finds the difference d between $I(x,y)$ and $\hat{I}(x,y)$ as: $d=\left|I(x,y)-\hat{I}(x,y)\right|$. It classifies d into four cases: $\lceil T/2 \rceil \le d < T$, $T \le d < T+\lfloor T/2 \rfloor$, $d \ge T+\lfloor T/2 \rfloor$ and $0 \le d < \lceil T/2 \rceil$, where T is a distortion control parameter. It calculates stego image values $I'(x,y)$ for non-reference pixels as follows: If d in the first case, insert a secret bit b as: $I'(x,y)=\hat{I}(x,y)+2\times d+b,\ if\ I(x,y)>\hat{I}(x,y)$ or $I'=\hat{I}(x,y)-2\times d-b, otherwise$.

In other cases, do not embed any secret bits. Find the stego pixel as:

$$I'(x,y)=\begin{cases} I(x,y)-\lfloor T/2 \rfloor,\ if\ I(x,y)>\hat{I}(x,y) \\ I(x,y)+\lfloor T/2 \rfloor, otherwise \end{cases} \quad \text{if } d \text{ in the second case,}$$

$$I'(x,y)=\begin{cases} I(x,y)+\lceil T/2 \rceil,\ if\ I(x,y)>\hat{I}(x,y) \\ I(x,y)-\lceil T/2 \rceil, otherwise \end{cases} \quad \text{if } d \text{ in the third case}$$

and $I'(x,y)=I(x,y)$ if d in the fourth case.

For extraction, scan the stego image I' in the same embedding raster scan order and find the prediction values as: $\hat{I}'(x,y)=\lfloor (I'(x-1,y)+I'(x,y-1))/2 \rfloor$. Next, find differences $d'=\left|I'(x,y)-\hat{I}'(x,y)\right|$ and classify d' into four cases: $T \le d < 2\times T$, $\lceil T/2 \rceil \le d' < T$, $d' \ge 2\times T$ and $0 \le d' < \lceil T/2 \rceil$. In the first case, extract a secret bit b and restore the cover pixel $I(x,y)$ as:

$$b=d'-\lfloor d'/2 \rfloor \text{ and } I(x,y)=\begin{cases} I'(x,y)+\lfloor T/2 \rfloor,\ if\ I'(x,y)>\hat{I}'(x,y) \\ I'(x,y)-\lfloor T/2 \rfloor, otherwise \end{cases} .$$

For other cases: Do not extract any secret bits, restore the cover pixel as:

$$I(x,y)=\begin{cases} I'(x,y)+\lfloor T/2 \rfloor,\ if\ I'(x,y)>\hat{I}'(x,y) \\ I'(x,y)-\lfloor T/2 \rfloor, otherwise \end{cases} , \ d' \text{ in the second case.}$$

$$I(x,y)=\begin{cases} I'(x,y)-\lceil T/2 \rceil,\ if\ I'(x,y)>\hat{I}'(x,y) \\ I'(x,y)+\lceil T/2 \rceil, otherwise \end{cases} , \ d' \text{ in the third case.}$$

$I'(x,y)=I(x,y)$, d' in the fourth case.

This scheme does not require a location map to recover the original cover image. However, the number of pixels in case one limits its capacity.

2.2. Chen-Feng et al. Scheme

Given a cover image I , let $I(x,y)$ denote the gray level of the pixel with coordinates (x,y). Let T be a distortion control parameter and define a shrinkage parameter $\delta=T+1$. The first step is a preprocessing step to prevent overflow and underflow problems. In this step, replace boundary pixels in the range $[0,\delta]$ by δ and replace values in the range $[\,255-\delta, 255\,]$

by [$255-\delta$]. The second step finds the prediction value $\hat{I}(x,y)$ for all pixels except for the first row and column of I . The prediction process proceeds in a raster scan from left to right and top to bottom. Calculate the prediction value as: $\hat{I}(x,y)=\lfloor (I(x-1,y)+I(x,y-1))/2 \rfloor$. Find the difference d between $I(x,y)$ and $\hat{I}(x,y)$ as $d=|I(x,y)-\hat{I}(x,y)|$. Classify d into two cases: $d \le T$ and $d > T$. Calculate stego image values $I'(x,y)$ for non-reference pixels and insert a secret bit b as follows:

Case 1: $d \le T$, insert a secret bit b as: $I'(x,y)=\begin{cases}\hat{I}(x,y)+2\times d+b, \ if\ I(x,y)\ge \hat{I}(x,y)\\ \hat{I}(x,y)-2\times d-b, otherwise\end{cases}$.

Case 2: $d > T$, modify the pixel as: $I'(x,y)=\begin{cases}I(x,y)+\delta, \ if\ I(x,y)\ge \hat{I}(x,y)\\ I(x,y)-\delta, otherwise\end{cases}$.

For extraction and recovery, scan stego image $I'(x,y)$ in the same embedding order and find the prediction values for non-reference pixels as:

$\hat{I}'(x,y)=\lfloor (I'(x-1,y)+I'(x,y-1))/2 \rfloor$ and find differences $d'=|I'(x,y)-\hat{I}'(x,y)|$. Then classify d' into two cases:

Case 1: $d' \le 2\times T+1$, extract a secret bit b and restore cover pixel $I(x,y)$ as:

$b=LSB(d')$ and $I(x,y)=\begin{cases}\hat{I}'(x,y)+\lfloor d'/2 \rfloor, \ if\ I'(x,y)\ge \hat{I}'(x,y)\\ \hat{I}'(x,y)-\lfloor d'/2 \rfloor, otherwise\end{cases}$.

Case 2: $d' > 2\times T+1$, restore cover pixel $I(x,y)$ as: $I(x,y)=\begin{cases}I'(x,y)-\delta, \ if\ I'(x,y) > \hat{I}'(x,y)\\ I'(x,y)+\delta, otherwise\end{cases}$.

3. THE PROPOSED SCHEME

In this section, we present a new RDHT based on pixel prediction and difference expansion. The proposed scheme consists of three steps: Image preprocessing step, embedding step and extraction step. Both embedding and extraction steps need to predict pixels' values.

3.1. Preprocessing Procedure

The objective of this step is to prevent overflow and underflow problems in the embedding step. Given a gray level cover image I of size $n\times m$, let $p(x,y)$ denote the gray level of the pixel with coordinates (x,y). Let T be a distortion control parameter and define a shrinkage parameter $\delta = T+1$. In this step, replace boundary non-reference pixels in the range [$0,\delta$] by δ and replace values in the range [$255-\delta, 255$] by [$255-\delta$]. The effect of this step is to shrink the histogram of non-reference pixels from both ends. Fig. 1 shows that the values of the modified pixels can be recovered without the need for a location as will be shown in subsection 3.4.

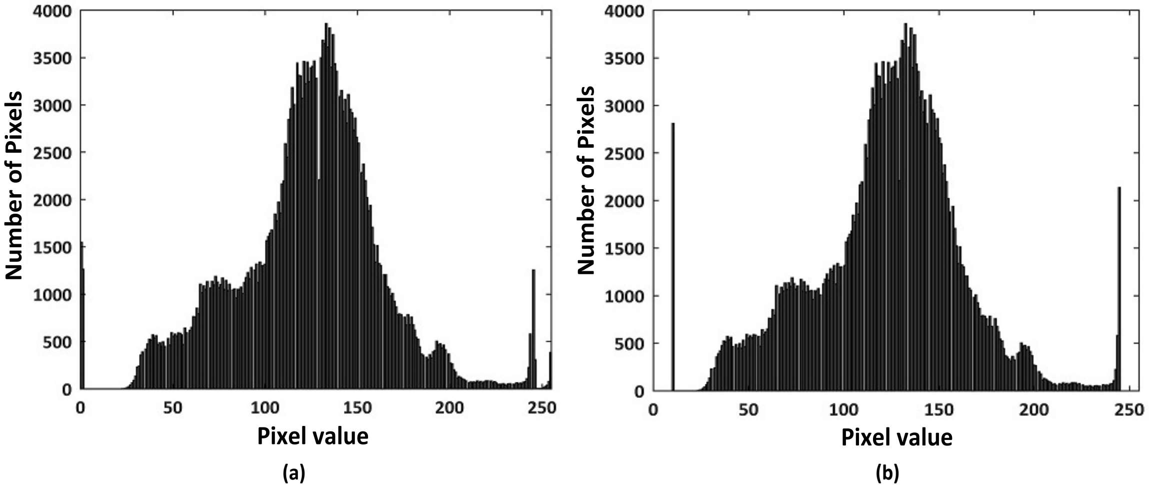


Fig. 1. Image histogram: a) original; b) after shrinkage.

3.2. Pixel Prediction Procedure

Given the preprocessed cover image I, divide it into 3×3 overlapping blocks B_k, where $k = 1,2,3,..., \lfloor (n-1)/2 \rfloor \times \lfloor (m-1)/2 \rfloor$. The center coordinates of these blocks are $(2\times i, 2\times j)$, $i = 1,2,3,..., \lceil n/2 \rceil - 1$ and $j = 1,2,3,..., \lceil m/2 \rceil - 1$. Fig. 2 shows the block division for 5×5 and 6×6 images. In Fig. 2, we denote the center pixels of these blocks by: C_1, C_2, C_3 and C_4. In case n and/or m are even, the last row and/or column will not be a part of the 3×3 blocks' region as indicated by the dark shaded pixels in Fig. 2(b). We divide the prediction algorithm into two parts. The first deals with the 3×3 blocks while the second deals with the last row and/or column in case of n and/or m are even.

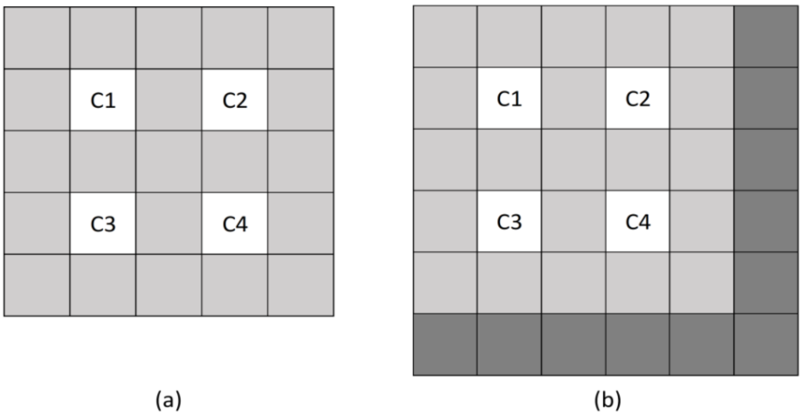


Fig. 2. Block division: a) 5x5 image; b) 6x6 image.

3.2.1. Prediction Algorithm for 3×3 Blocks

This algorithm is applied to all 3×3 blocks defined above in a raster scan order from right to left and top to bottom. Fig. 3(a) shows an original 3×3 block while Fig. 3(b) shows the predicted block. We classify its pixels into two classes. Class 1 is composed of reference pixels that will carry no secret data and remain intact during the embedding process. The corner pixels in Fig. 3 represent this class. Class 2 is composed of the remaining bits of the block. Pixels in this class may carry secret bits.

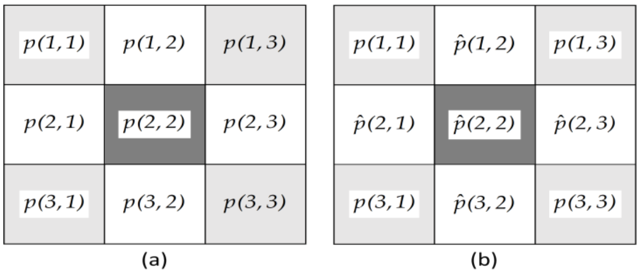


Fig. 3. 3×3 block B_k: a) original; b) predicted.

The proposed prediction algorithm for the 3×3 block finds predicted values of embedding pixels using Eq. (1):

$$\begin{aligned}
&\hat{p}(2,2)=\lfloor (p(1,1)+p(1,3)+p(3,1)\ +\ p(3,3))/4\rfloor\\
&\hat{p}(1,2)=\lfloor (p(1,1)+p(1,3)+p(2,2))/3\rfloor\\
&\hat{p}(2,1)=\lfloor (p(1,1)+p(3,1)+p(2,2))/3\rfloor\\
&\hat{p}(2,3)=\begin{cases}\lfloor (p(1,3)+p(3,3)+p(2,2))/3\rfloor, \text{ if in last block in a row}\\ p(2,3)\ ,\ \text{otherwise}\end{cases}\\
&\hat{p}(3,2)=\begin{cases}\lfloor (p(3,1)+p(3,3)+p(2,2))/3\rfloor, \text{ if in last block in a column}\\ p(3,2)\ ,\ \text{otherwise}\end{cases}
\end{aligned}\tag{1}$$

3.2.2. *Prediction Algorithm for the Last Row and Column*

In the case of cover image with an even number of rows and/or columns, the last row and/or column are not included in the 3×3 blocks' region and can't be predicted using Eqs. (1) and (2). The prediction algorithm for the last row and column starts by assigning reference pixels $p(2q,m)$ and $p(n,2r)$, where $q=0,1,2,....,n/2$ and $r=0,1,2,....,m/2$ as shown in Fig. 4. Eq. (2) finds predicted pixel values of the last row and column:

$$\begin{aligned}
&\hat{p}(1,m)=\lfloor (p(1,m-1)+p(2,m-1)+p(2,m))/3\rfloor\\
&\hat{p}(i,m)=\lfloor (p(i,m-1)+p(i-1,m)+p(i+1,m))/3\rfloor\ ,\ i=3,5,7\ ,...,n-1\\
&\hat{p}(n,1)=\lfloor (p(n-1,1)+p(n-1,2)+p(n,2))/3\rfloor\\
&\hat{p}(n,j)=\lfloor (p(n-1,j)+p(n,j-1)+p(n,j+1))/3\rfloor\ ,\ j=3,\ 5,7,...,m-1
\end{aligned}\tag{2}$$

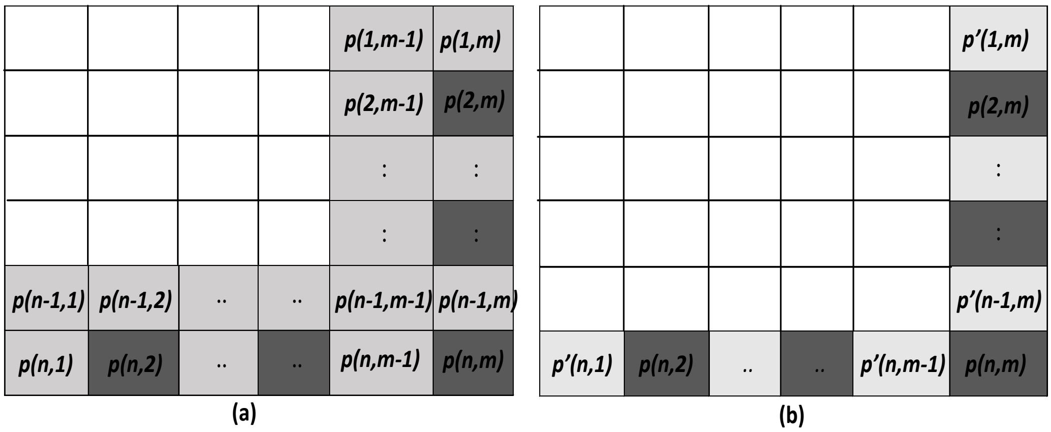


Fig. 4. Prediction of the last row and column: a) original image $I(x,y)$; b) predicted image $I'(x,y)$.

The following example gives an illustration of the prediction algorithm. Fig. 5 shows an original 4×4 sample image and its predicted image. In this example, there is only one 3×3 block with center coordinates $(2,2)$. Eqs. (1) and (2) give its predicted values as:

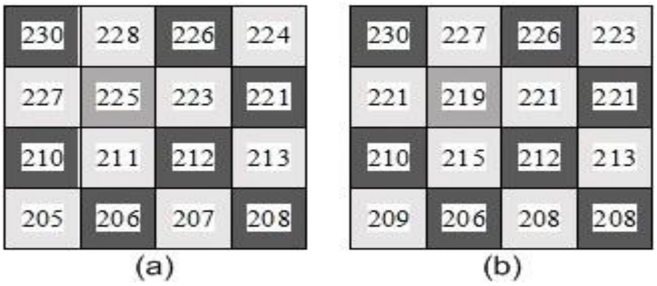
$\hat{p}(2,2)=\lfloor(230+226+210+212)/4\rfloor=219$, $\hat{p}(1,2)=\lfloor(230+226+225)/3\rfloor=227$,

$\hat{p}(2,1)=\lfloor(230+227+225)/3\rfloor=221$, $\hat{p}(3,1)=\lfloor(210+225+212)/3\rfloor=215$,

$p(2,3)=\lfloor(226+212+225)/3\rfloor=221$, $\hat{p}(1,4)=\lfloor(223+221+226)/3\rfloor=223$,

$\hat{p}(3,4)=\lfloor(221+212+208)/3\rfloor=213$, $\hat{p}(4,1)=\lfloor(206+211+210)/3\rfloor=209$ and

$\hat{p}(4,3)=\lfloor(206+212+208/3\rfloor=208$.



230	228	226	224
227	225	223	221
210	211	212	213
205	206	207	208

(a)

230	227	226	223
221	219	221	221
210	215	212	213
209	206	208	208

(b)

Fig. 5. Prediction example: a) original image I ; b) predicted image $\hat{I}$.

3.3. Embedding Procedure

Given the original cover image $I(x,y)$, the embedding procedure consists of the following steps:

Step 1: Except for reference-pixels, preprocess the cover image by shifting border pixels using Eq. (3):

$$I(x,y)=\begin{cases}\delta, & if\ \ 0\le I(x,)<\delta\\ 255-\delta, & if\ \ I(x,)>255-\delta\end{cases}\tag{3}$$

Step 2: Assign reference pixels and carry out block division as outlined in subsection 3.1.

Step 3: Following the same sequence outlined in subsection 3.2., find predicted pixel-values $\hat{p}(x,y)$ for each embeddable pixel $p(x,y)$ using Eqs. (1) and (2).

Step 4: Using the same sequence above, find the difference $d=|(p(x,y)-\hat{p}(x,y)|$. Then, to embed a secret bit s , we have two cases:

Case 1: $d\le T$, embed a secret bit s and find the stego pixel $p'(x,y)$ using Eq. (4):

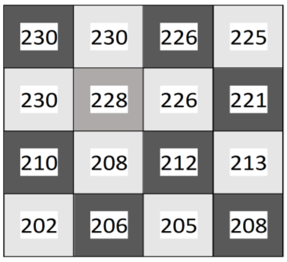
$$p'(x,y)=\begin{cases}\hat{p}(x,y)+2d+s, & if\ \hat{p}(x,y)\le p(x,y)\\ \hat{p}(x,y)-2d-s, & otherwise\end{cases}\tag{4}$$

Case 2: $d>T$, do not embed secret bit and find stego-pixel $p'(x,y)$ using Eq. (5):

$$p'(x,y)=\begin{cases}p(x,y)+\delta, & if\ \hat{p}(x,y)\le p(x,y)\\ p(x,y)-\delta, & otherwise\end{cases}\tag{5}$$

As an example of the embedding procedure, we carry out the embedding steps to the cover image in Fig. 5(a). We have calculated predicted pixel values using Eqs. (1) and (2) in subsection 3.2. To find the stego image, let $T=2$, $\delta=T+1=3$ and the secret message $S=$ '11001'. Start with the upper left 3×3 block shown in Fig. 6(a). Starting with upper left

3×3 block, the center-pixel $p(2,2)=225$, $\hat{p}(2,2)=219$ with a difference $d=|225-219|=6$. Since $d>T$, no secret bit is inserted and the stego pixel value is obtained using Eq. (5) as $p'(2,2)=p(2,2)+\delta=225+3=228$. Pixel $p(1,2)=228$ with predicted value $\hat{p}(1,2)=227$ and $d=|228-227|=1$. Since $d\le T$, the first secret bit $s=$'1' is inserted and the stego pixel value is obtained using Eq. (4) as: $p'(1,2)=\hat{p}(1,2)+2d+s=227+2\times1+1=230$. Pixel $p(2,1)=227$, its predicted value is $\hat{p}(2,1)=221$ and $d=|227-221|=6$. Since $d>T$, no secret bit is inserted and the stego pixel value is obtained using Eq. (5) as: $p'(2,1)=227+3=230$. Pixel $p(2,3)=223$, with predicted value $\hat{p}(2,3)=221$ and $d=|223-221|=2$. Since $d\le T$, the second secret bit $s=$'1' is inserted and the stego pixel value is obtained using Eq. (4) as: $p'(2,3)=\hat{p}(2,3)+2d+s$ $=221+2\times2+1=226$. Pixel $p(3,2)=211$ with predicted value $\hat{p}(3,2)=215$ and $d=|211-215|=4$. Since $d>T$, no secret bit is inserted and the stego pixel value is obtained using Eq. (5) as: $p'(3,2)=211-3=208$. We then proceed to the final column. Pixel $p(1,4)=224$, $\hat{p}(1,4)=223$ and $d=|224-223|=1$. Since $d\le T$, the third secret bit $s=$'0' is inserted and the stego pixel value is obtained using Eq. (4) as: $p'(1,4)=\hat{p}(1,4)+2d+s=222+2\times1+0=225$. Pixel $p(3,4)=213$ with $\hat{p}(3,4)=213$ and $d=|213-213|=0$. Since $d\le T$, the fourth secret bit $s=$'0' is inserted and the stego pixel value is obtained using Eq. (4) as: $p'(3,4)=\hat{p}(3,4)+2d+s=213+2\times0+0=213$. Finally we come to the final row. Pixel $p(4,1)=205$ with predicted value $\hat{p}(4,1)=209$ and $d=|205-209|=4$. Since $d>T$, no secrete bit is inserted and the stego pixel value is obtained using Eq. (6) as $p'(4,1)=205-3=202$. Pixel $p(4,3)=207$ with $\hat{p}(4,3)=208$ and $d=|207-208|=1$. Since $d\le T$, the fifth secret bit $s=$'1' is inserted and the stego pixel value is obtained as $p'(4,3)=\hat{p}(4,3)-2d-s=208-2\times1-1=205$. Fig. 6 shows the resulting stego image $I'(x,y)$.



230	230	226	225
230	228	226	221
210	208	212	213
202	206	205	208

Fig. 6. Embedding example of stego-image I'.

3.4. Data Extraction and Image Recovery Procedure

Given the stego-image I', follow the same procedure in subsection 3.2 to divide it into 3×3 blocks B'_k and assign reference pixels. The extraction and image recovery starts by scanning 3×3 blocks B'_k region then, respectively, proceed to the last column n and row m in the case of even n and m. Then for each embedded pixel $p'(x,y)$, the extraction and recovery procedure proceeds as follows:

Step 1: Find the predicted value $\hat{p}'(x,y)$ using Eq. (1) if the pixel is within a block or use Eq. (2) if the pixel is in the last row or column in the case of even n and m.

Step 2: Find the distance d' using Eq. (6) as:

$$d'=|p'(x,y)-\hat{p}'(x,y)| \tag{6}$$

Step 3: If $d' \le 2T+1$ then,
Extract the secret bit s and recover the original pixel $I(x,y)$ using Eq. (7) as:

$$s = \mathrm{mod}(d',2)$$

$$p(x,y) = \begin{cases} \hat{p}'(x,y) + \lfloor d'/2 \rfloor, & if\ \hat{p}'(x,y) \le p'(x,y) \\ \hat{p}'(x,y) - \lfloor d'/2 \rfloor, & otherwise \end{cases} \tag{7}$$

else No secret bit is extracted
Recover the original pixel using Eq. (8) as:

$$p(x,y) = \begin{cases} p'(x,y) - \delta, & if\ \hat{p}'(x,y) \le p'(x,y) \\ p'(x,y) + \delta, & otherwise \end{cases} \tag{8}$$

The following example details the extraction and image-recovery procedure. Fig. 6 shows the stego-image $I'(x,y)$ that resulted from the embedding example with $T=2$, $\delta = T+1 = 3$ and the secret message $S =$ $'11001'$. Following the embedding sequence, we start with the 3×3block. The center pixel $p'(2,2)=228$ and using Eq. (1), the predicted value is $\hat{p}'(2,2) = \lfloor (230+226+212+210)/4 \rfloor = 219$ with a difference $d' = |228-219| = 9$. Since $d' > 2T+1$, no secret bit is extracted and the cover pixel value is obtained using Eq. (8) as $p(2,2) = p'(2,2) - \delta \ = 228-3 = 225$. Pixel $p'(1,2)=230$ with predicted value $\hat{p}'(1,2) =$ $\lfloor (230+226+225)/3 \rfloor \ = 227$ and $d' = |230-227| = 3$. Since $d' \le 2T+1$, the first secret bit is extracted and the cover pixel value is obtained using Eq. (7) as: $s = \mathrm{mod}(d',2) \ = '1'$ and the cover pixel $p(1,2) = \hat{p}'(1,2) + \lfloor d'/2 \rfloor \ = 227+1 = 228$. The next pixel is $p'(2,1)=230$ with $\hat{p}'(2,1) = \lfloor (230+210+225)/3 \rfloor = 221$ and $d' = |230-221| = 9$. Since $d' > 2T+1$, no secret bit is extracted and the cover pixel value is obtained using Eq. (8) as $p(2,1) = 230-3 = 227$. Pixel $p'(2,3) = 226$ comes next with $\hat{p}'(2,3) = \lfloor (226+212+225)/3 \rfloor = 221$and $d' = |226-221| = 5$. Since $d' \le 2T+1$, the second secret bit extracted as $s = \mathrm{mod}(5,2) = '1'$ and the cover pixel value is obtained Eq. (7) as $p(2,3) = \hat{p}'(2,3) + \lfloor d'/2 \rfloor \ = 221 + \lfloor 5/2 \rfloor 223$. The final embedded pixel in this block is $p'(3,2) = 228$, $\hat{p}'(3,2) = \lfloor (210+212+225)/3 \rfloor = 225$ and $d' = |208-215| = 7$. Since $d' > 2T+1$, no secret bit is extracted and the cover pixel value is obtained using Eq. (8) as $p(3,2) = 208+3 = 211$. In the final column, pixel $p'(1,4) = 225$, $\hat{p}'(1,4) = \lfloor (226+223+221)/3 \rfloor = 223$ and $d' = |225-223| = 2$. Since $d' \le 2T+1$, the third secret bit is extracted as $s = \mathrm{mod}(2,2) = '0'$ and the cover pixel value is obtained using Eq. (7) as $p(1,4) = 223 + \lfloor 2/2 \rfloor = 224$. Pixel $p'(3,4) = 213$ with, $\hat{p}(3,4) = \lfloor (221+212+208)/3 \rfloor = 213$ and $d' = |213-213| = 0$. Since $d' \le 2T+1$, the fourth secret bit is extracted as $s = \mathrm{mod}(0,2) = '0'$ and the cover pixel value is obtained using Eq. (7) as $p(3,4) = \hat{p}'(3,4) + \lfloor d'/2 \rfloor = 213 + \lfloor 0/2 \rfloor = 213$. Pixel $p'(4,1) = 202$ with predicted value $\hat{p}(4,1) = \lfloor (210+211+206)/3 \rfloor = 209$ and $d' = |209-202| = 7$. Since $d' > 2T+1$, no secret bit is extracted and the cover pixel value is obtained using Eq. (8) as $p(4,1) = 202+3 = 205$. The final embeddable pixel in this row is $p'(4,3) = 205$ with $\hat{p}'(4,3) =$ $\lfloor (206+212+208)/3 \rfloor = 208$ and $d' = |205-208| = 3$. Since $d' \le 2T+1$, the fifth secret bit is extracted as $s = \mathrm{mod}(3,2) = '1'$ and the stego pixel value is obtained using Eq. (7) as $p(4,3) = \hat{p}'(4,3) - \lfloor d'/2 \rfloor = 208 - \lfloor 3/2 \rfloor = 207$.

3.5. Recovery of Extreme Pixel Values

As was shown in subsection 3.1, except for reference pixels, extreme pixel values have to be adjusted before embedding to avoid overflow and underflow problems during the embedding process. Pixel values in the range $0 \le p(x,y) \le \delta$ are set to $p(x,y) = 255 - \delta$ and pixel values in the range $255-\delta \le p(x,y) \le 255$ are set to $p(x,y) = 255 - \delta$. In the extraction and recovery step, recovered pixel values of $p(x,y) = \delta$ or $p(x,y) = 255 - \delta$ indicate the location of adjusted pixels. Thus, there is no need for a location map to recover these pixels. However, we need to recover the original pixel values. To do this, we decode each pixel value of these pixels into an nn -bit string, $nn = \lceil \log_2(\delta + 1) \rceil$ to decide the position of the pixel value in the range $[0, \delta]$ or $[255 - \delta, 255]$. For example, if $\delta = 3$, then $nn = 2$ and the original pixel values 0 and 252 are decoded into two bits string $'00'$. Original pixel values 1 and 253 are decoded into two bit string $'01'$. Original pixel values 2 and 254 are decoded into two bit string $'10'$. Original pixel values 3 and 255 are decoded into two bit string $'11'$. We sequentially concatenate the codes for all of these pixels to form the overhead information. Then overhead information is appended to the left of the secret message forming the payload bit string. At the receiving end, overhead information will be extracted first. Whenever a recovered pixel value of δ or $255 - \delta$ is found, two bits from the overhead stream are sequentially used to recover the original pixel value before preprocessing. For example, if $\delta = 3$ and the first two bits of the extracted overhead information is $'10'$, then if the recovered pixel value is $p(x,y) = \delta = 3$ and the original pixel value before preprocessing is 2. If the recovered pixel value is 252, then the original pixel value before preprocessing is $p(x,y) = 254$.

4. EXPERIMENTAL RESULTS

This section presents the simulation results of the proposed RDHT in terms of peak signal to noise ratio (PSNR) values and data hiding capacity.

4.1. Experimental Setup

For the distortion measure, we used the PSNR between the $M \times N$ original cover image I and the stego image I'. PSNR is given by:

$$PSNR(I, I') = 10 \ \log_{10} \frac{M \times N \times (255)^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} (X(i,j) - I'(i,j))^2} \ \text{dB} \ .$$

The overhead (OH) is given by:

$$OH = \text{Number of shifted bits in preprocessing step} \times \log_2(\delta + 1) \ .$$

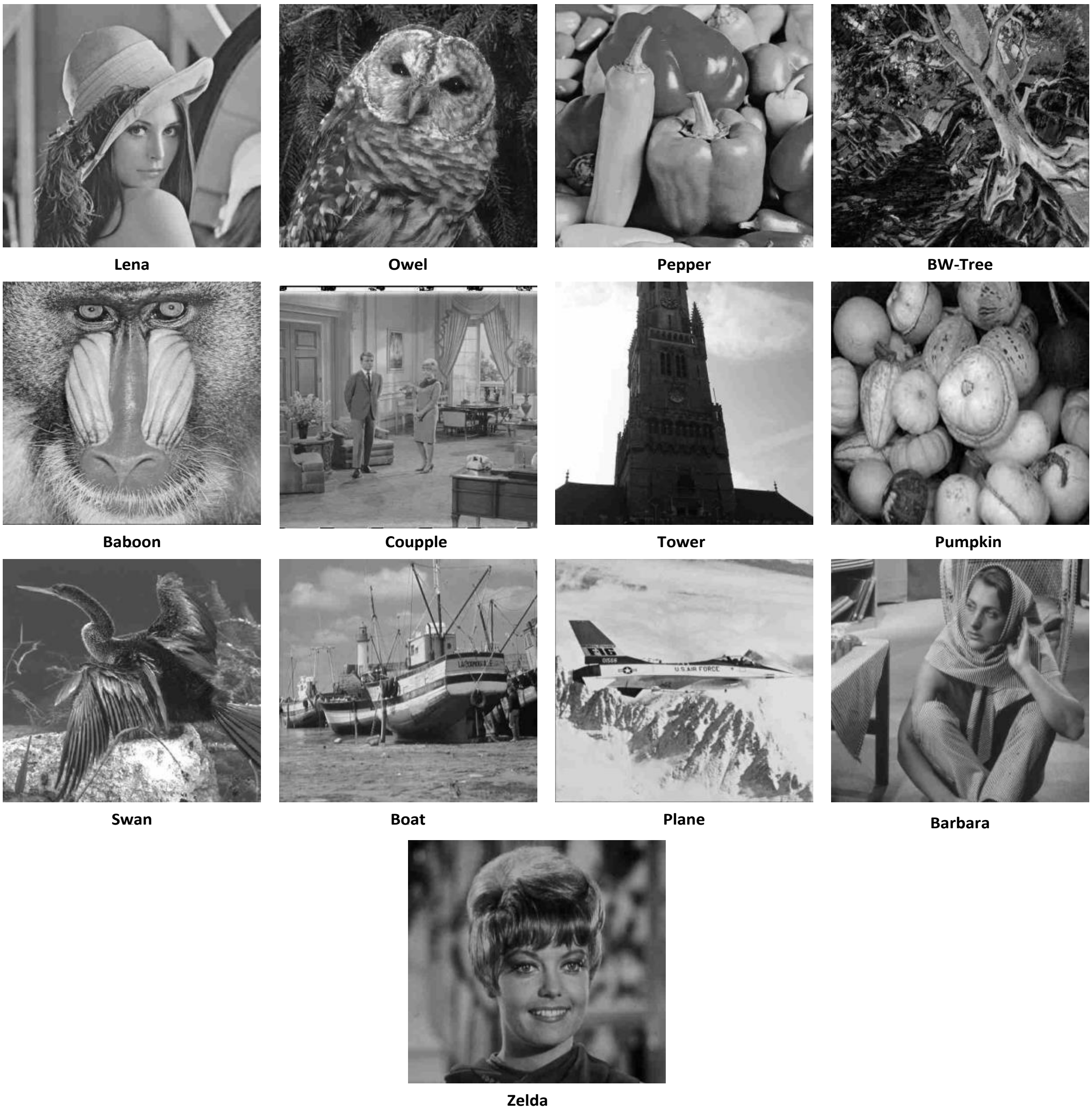
The data hiding capacity (H) is the difference between the total number of embedded bits and the overhead bits:

$$H = \text{total number of embedde bits - } OH \ .$$

We also define the embedding rate (ER)as:

$$ER = H / (M \times N) \ .$$

In our simulations, we used thirteen 512×512 grayscale cover images. Fig. 7 shows these test images.



Lena Owel Pepper BW-Tree

Baboon Coupple Tower Pumpkin

Swan Boat Plane Barbara

Zelda

Fig. 7. The utilized test images

4.2. Performance of the Proposed RDHT Scheme

The distortion control parameter (T) determines the visual quality and embedding capacity of the proposed scheme. Smaller values of T results in high PSNR values and lower embedding capacity while larger values of T results in lower PSNR values and higher embedding capacity.

Fig. 8 shows the embedding capacity as a function of T. The capacity starts increasing as T increases up to a point where the overhead becomes high and reduces embedding capacity as shown by 'Tree' and 'Tower' plots. The average PSNR value at $T = 0$ is 49.7 dB with an average embedding capacity of 23,830 bits (0.09 bpp embedding rate). At $T = 1$, the average embedding capacity is 55,593 bits (0.21 bpp embedding rate) with an average PSNR of 44.28 dB. At $T = 2$, the average embedding capacity is 80,309 bits (0.31 bpp embedding rate) with an average PSNR of 41.42 dB. This demonstrates the effectiveness of the proposed scheme for small values of T.

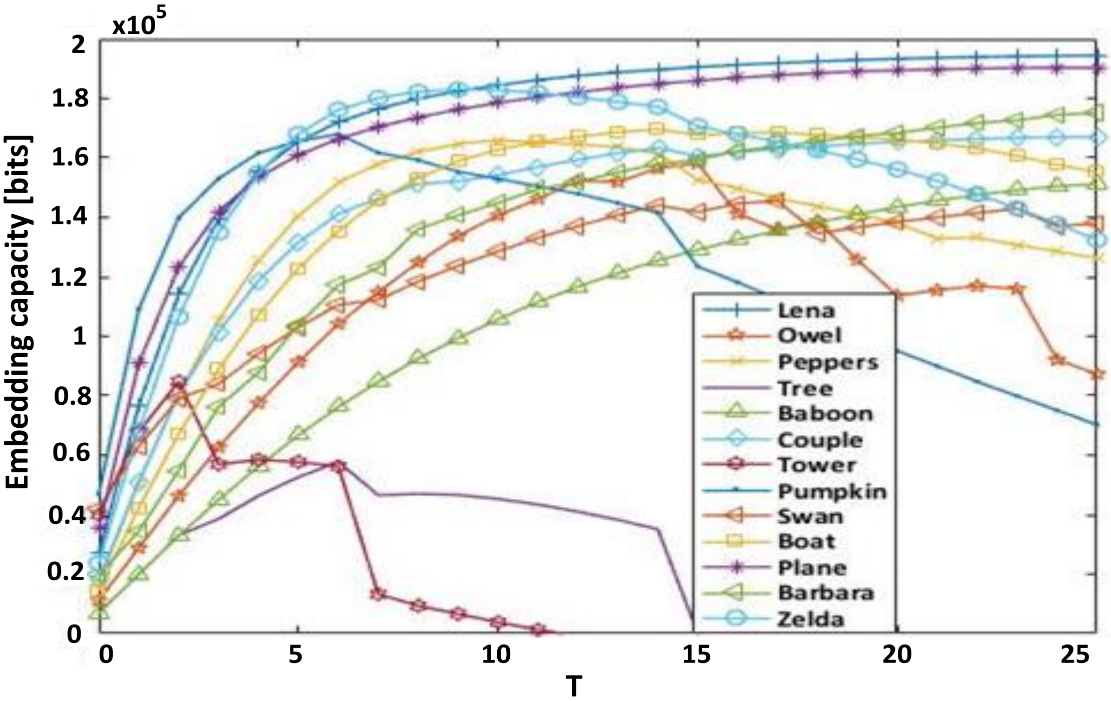


Fig. 8. Data hiding capacity as a function of *T*.

On the other side, for large values of T, Table 1 gives the maximum capacity with the corresponding PSNR values and the values of *T* that achieved it.

Table 1. Maximum capacity performance.

Image	T	PSNR [dB]	Max capacity [bits]	Embedding rate [bpp]
Lena	26	34.33	194,487	0.74
Owl	15	30.17	158,217	0.60
Peppers	10	34.58	165,938	0.63
BW-Tree	6	34.29	57,787	0.22
Baboon	30	24.95	152,945	0.58
Couple	25	31.17	167,123	0.64
Tower	2	43.10	84,795	0.32
Pumpkin	6	39.67	167,989	0.64
Swan	17	30.37	145,574	0.56
Boat	14	32.25	169,660	0.65
Plane	24	33.00	190,376	0.73
Barbara	26	27.84	175,975	0.67
Zelda	9	37.04	183,020	0.70

It is widely accepted that the minimum PSNR value for indiscernible distortion to the human eye is 30 dB. Except for 'Baboon' and 'Barbara', all images achieved maximum embedding rate with PSNR values higher than 30 dB. The average maximum capacity is 154,914 bits (0.59 bpp embedding rate) with 33.29 dB average PSNR. This demonstrates the effectiveness of the proposed scheme for large values of *T*.

Fig. 9 shows the plots of PSNR vs. embedding rate for values of T = 0 up to T = 30. It is clear that the proposed technique performance can be adjusted via the choice of T to obtain the desired stego-image quality with PSNR values up to 49 dB at low embedding rates. Even at high embedding rates, the proposed technique results in relatively high PSNR values. For example, at 0.74 bpp embedding rate, PSNR value is 34.33 dB in 'Lena image'.

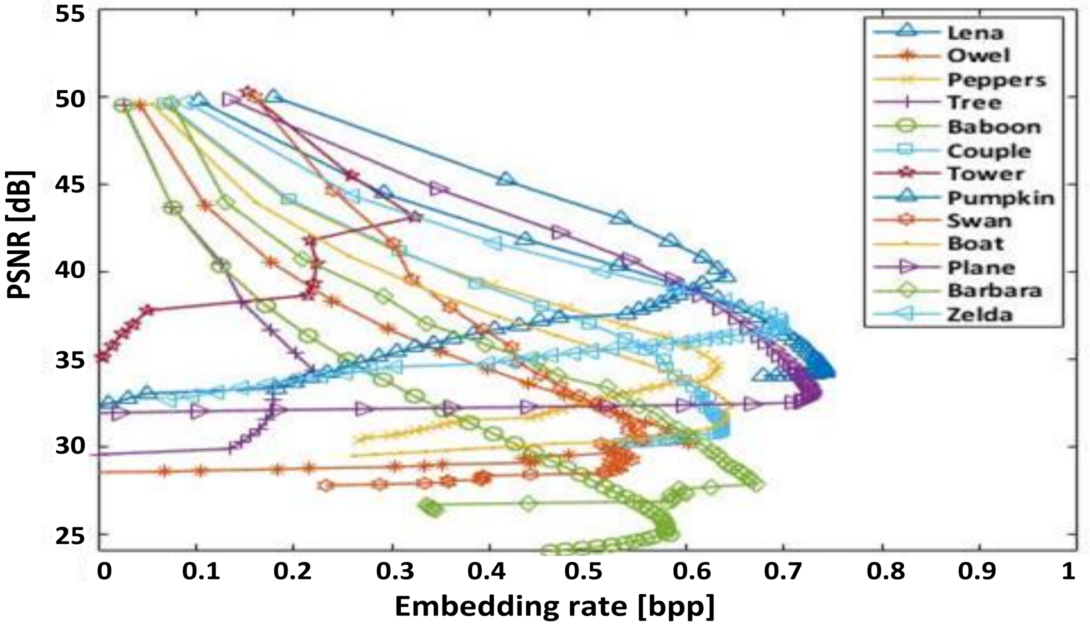


Fig. 9. PSNR vs. embedding rate.

Fig. 10 compares stego-image quality for 'Lena' image at different embedding rates. It is clear that even at maximum embedding rate; no image distortion can be noticed. This is true for all test images. As for fidelity, we have shown in the extraction procedure that the original cover image is exactly recovered after the extraction of the secret message. In summary, by the adjustment of T , the proposed scheme gives an adjustable RDHT with real high-quality stego images and medium embedding capacity.

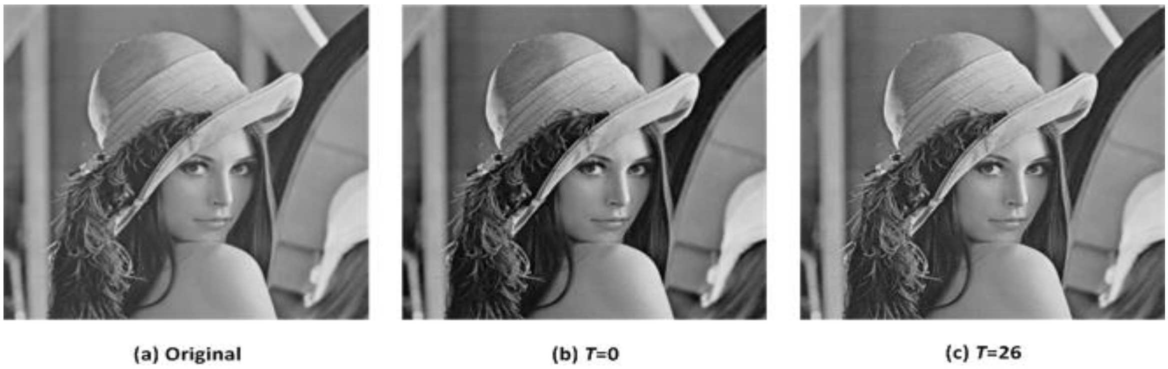


Fig. 10. Lena image: a) original; b) embedded 0.74 bpp; c) embedded 0.1 bpp.

4.3. Comparison with Related Work

This section compares the performance of the proposed technique with closely related techniques of Chen-Fing et al. [4] and Tseng and Hsieh [15]. Table 2 summarizes the maximum embedding rate performance, the PSNR values and the distortion control parameter T that gives the maximum embedding rate for the thirteen test images. In terms of embedding rate, the proposed technique gives more than twice the average embedding rate of Tseng and Hsieh and about 78% of Chen et al. In terms of PSNR values, the proposed algorithm outperforms Chen et al. by an average of 2.33 dB. In eleven out of the thirteen test images, PSNR values of the proposed algorithm are above the 30 dB threshold mentioned in the previous section. On the other hand, Chen et al. technique gives only seven out of thirteen PSNR values above the 30 dB threshold.

Table 2. Maximum capacity performance comparisons.

Image	The proposed technique			Tsing and Hseih [15]			Chen-Fing et al. [4]		
	T	PSNR [dB]	Embedding rate [bpp]	T	PSNR [dB]	Embedding rate [bpp]	T	PSNR [dB]	Embedding rate [bpp]
Lena	26	34.33	0.74	4	41.11	0.30	23	31.09	0.93
Owl	15	30.17	0.60	10	33.90	0.23	15	23.26	0.77
Peppers	10	34.53	0.63	6	37.92	0.30	11	31.39	0.31
BW-Tree	6	34.29	0.22	2	47.37	0.05	6	32.73	0.26
Baboon	30	24.95	0.53	14	31.42	0.23	30	23.33	0.76
Couple	25	31.17	0.64	6	33.31	0.24	27	23.93	0.35
Tower	2	43.10	0.32	-	-	-	2	41.40	0.46
Pumpkin	6	39.67	0.64	2	46.05	0.24	6	36.03	0.79
Swan	17	30.37	0.56	14	32.73	0.07	17	23.56	0.72
Boat	14	32.25	0.65	3	35.95	0.23	17	29.41	0.34
Plane	24	33.00	0.73	2	46.00	0.27	24	30.59	0.95
Barbara	26	27.34	0.67	6	37.43	0.29	25	26.11	0.33
Zelda	9	37.04	0.70	6	33.36	0.32	10	34.06	0.91
Average	-	33.29	0.59	-	33.39	0.24	-	30.96	0.77

Figs. 11 and 12 give comparison examples of PSNR vs. embedding rate performance. It is clear that the proposed algorithm gives better PSNR values for embedding rates below 0.65 bpp with a minimum of 34 dB at 0.74 bpp embedding rate.

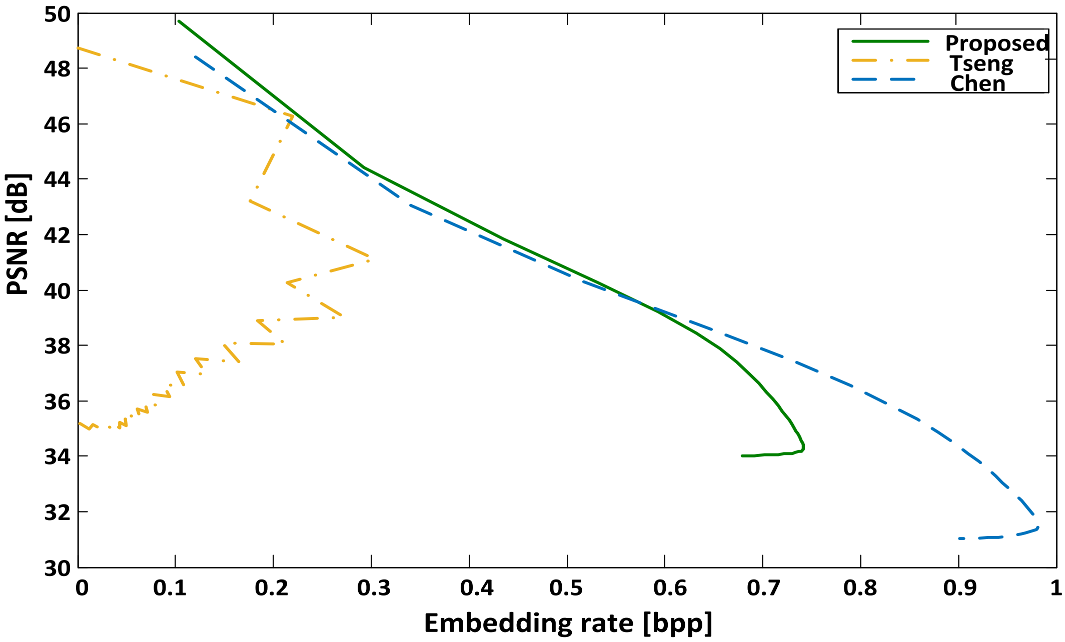


Fig. 11. PSNR vs. embedding rate (Lena image).

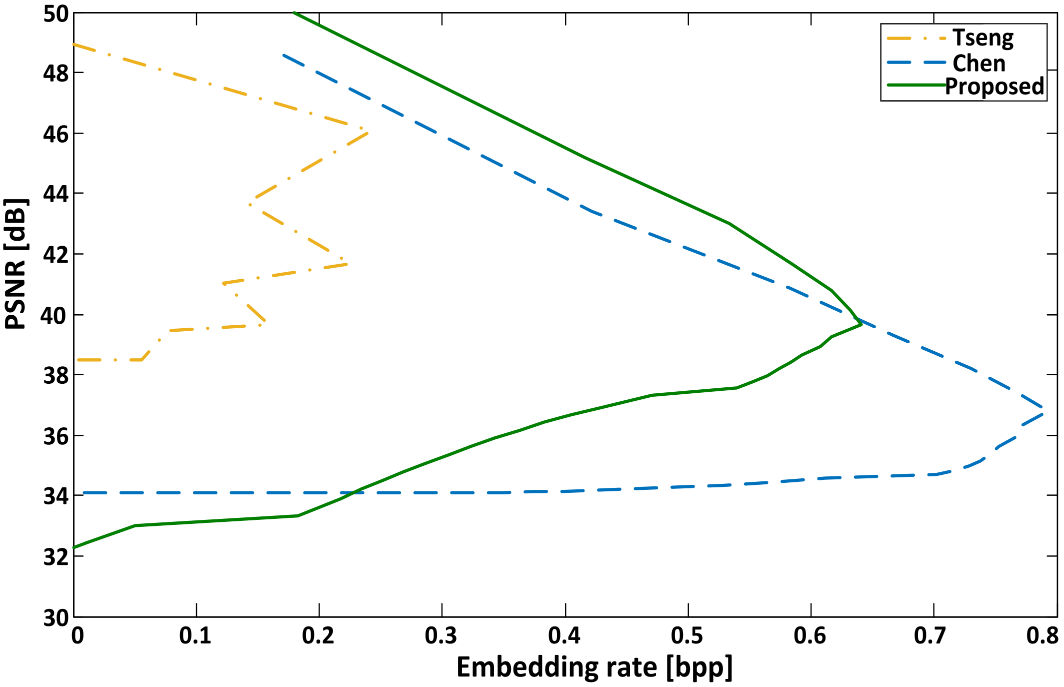


Fig. 12. PSNR vs. embedding rate (Pumpkin image).

It is clear that the proposed algorithm gives better PSNR values for embedding rates below 0.65 bpp with a minimum of 34 dB at 0.74 bpp embedding rate.

5. CONCLUSIONS

In this paper, we proposed a new RDHT based on prediction-error and difference expansion to increase stego image quality and transparency. The proposed technique reduces image distortion via a new prediction procedure. The increased number of reference pixels (one-fourth of image size) and its distribution in the cover image reduces image distortion. The use of three or four adjacent pixels for interpolation reduces prediction error and thus, increasing the number of pixels that can carry secret data. The proposed RDHT is adjustable via a distortion control parameter T that one can adjust to provide the desired image quality and data hiding capacity. The receiver only needs to know T to extract the secret message and fully recover the original cover image from the stego image.

Experimental results show that the proposed technique results in high-quality stego images with moderate embedding rates. Regardless of the embedding rate, the resulting stego image quality is high with no noticeable distortions in the stego image. At minimum embedding rates, the average PSNR obtained is 49.7 dB. At maximum embedding rates, the average PSNR obtained is 33.29 dB. Embedding rates ranged from an average of 0.09 bpp up to an average of 0.59 bpp.

REFERENCES

[1] A. Mohammad, A. Al-Haj, M. Farfoura, "An improved capacity data hiding technique based on image interpolation," *Multimedia Tools and Applications*, vol. 78, no. 6, pp. 7181–7205, 2018.

[2] L. Liu, T. Chen, S. Zhu, W. Hong, X. Si, "A reversible data hiding method using improved neighbor mean interpolation and random-block division," *Information Technology Journal*, vol. 13, no. 15, pp. 2374–2384, 2014.

[3] H. Kim, V. Sachnev, Y. Shi, J. Nam, H. Choo, "A novel difference expansion transform for reversible data embedding," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 456-465, 2008.

[4] C. Lee, H. Chen, H. Tso, "Embedding capacity raising in reversible data hiding based on prediction of difference expansion," *Journal of Systems and Software*, 2010, vol. 83, no. 10, pp. 1864-1872, 2010.

[5] K. Jung, D. Kim, Y. Kim, K. Yoo, "Improved reversible data hiding scheme using weighted matrix with overlapping," *Information*, vol. 20, no. 9A, pp. 6565-6573, 2017.

[6] R. Kumar, K. Jung, "Robust reversible data hiding scheme based on two-layer embedding strategy," *Information Sciences*, vol. 512, pp. 96–107, 2020.

[7] R. Rajkumar, A. Vasuki, "Reversible and robust image watermarking based on histogram shifting," *Cluster Computing*, vol. 22, no. 5, pp. 12313–12323, 2019.

[8] M. Kumar, S. Agrawal, "Reversible data hiding based on prediction error expansion using adjacent pixels," *Security and Communication Networks*, vol. 9, no. 16, pp. 3703–3712, 2016.

[9] A. Malik, S. Singh, R. Kumar, "Recovery based high capacity reversible data hiding scheme using even-odd embedding," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 15803-15827, 2018.

[10] I. Dragoi, D. Coltuc, "Adaptive pairing reversible watermarking," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2420-2422, 2016.

[11] S. Kim, X. Qu, V. Sachnev, H. Kim, "Skewed histogram shifting for reversible data hiding using a pair of extreme predictions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3236-3246, 2019.

[12] S. Subburam, S. Selvakumar, S. Geetha, "High performance reversible data hiding scheme through multilevel histogram modification in lifting integer wavelet transform," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 7071-7095, 2018.

[13] L. Huang, L. Tseng, M. Hwang, "A reversible data hiding method by histogram shifting in high quality medical images," *Journal of Systems and Software*, vol. 86, no. 3, pp. 716-727, 2013.

[14] V. Sachnev, H. Kim, J. Nam, S. Suresh, Y. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989-999, 2009.

[15] H. Tseng, C. Hsieh, "Prediction-based reversible data hiding," *Information Sciences*, vol. 179, no. 14, pp. 2460-2469, 2009.

[16] D. Coltuc, "Improved embedding for prediction-based reversible watermarking," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 873-882, 2011.

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

[18] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, "Reversible image watermarking using interpolation technique," *Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, 2010.

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

[20] B. Jana, "High payload reversible data hiding scheme using weighted matrix," *Optik*, vol. 127, no. 6 , pp. 3347-3358, 2016.

[21] L. Liu, T. Chen, S. Zhu, W. Hong, X. Si, "A reversible data hiding method using improved neighbor mean interpolation and random-block division," *Information Technology Journal*, vol. 13, no. 15, pp. 2347, 2014.

[22] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[23] B. Macq, "Lossless multiresolution transform for image authenticating watermarking," *2000 10th European Signal Processing Conference*, Tampere, Finland, pp. 1-4, 2000.

[24] J. Barton, "Method and apparatus for embedding authentication information within digital data," *U.S. Patent 5 646 997*, 1997.

[25] D. Thodi, J. Rodrı´guez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.

[26] C. Honsinger, P. Jones, M. Rabbani, J. Stoffel, "Lossless recovery of an original image containing embedded data," *U.S. Patent 6 278 791*, 2001.

[27] C. De Vleeschouwer, J. Delaigle, B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Transactions of Multimedia*, vol. 5, no. 1, pp. 97–105, 2003.

[28] J. Fridrich, M. Goljan, R. Du, "Invertible authentication," *Proceedings of SPIE Photonics West, Security and Watermarking of Multimedia Contents III*, vol. 3971, San Jose, California, pp. 197-208, 2001.

[29] M. Celik, G. Sharma, A. Tekalp, E. Saber, "Reversible data hiding," *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, pp. 157-160, 2002.

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

[31] A. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.

[32] D. Thodi, J. Rodriguez, "Prediction-error based reversible watermarking," *Proceedings of 2004 International Conference on Image Processing*, vol. 3, pp.1549–1552, 2005.

[33] L. Kamstra, H. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2082–2090, 2005.

[34] Y. Hu, H. Lee, J. Li, "DE-based reversible data hiding with improved overflow location map," *IEEE Transactions of Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 250–260, 2009.

[35] W Tai, C. Yeh, C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, 2009.

[36] W. Hong, T. Chen, "Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism," *Journal of Visual Communication and Image Representation*, vol. 22, no. 2, pp. 131–140, 2011.

[37] Z. Ni, Y. Shi, N. Ansari, W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp.354–362,2006.

[38] P. Tsai, Y. Hu, Y. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, vol. 89, no. 6, pp. 1129–1143, 2009.

[39] H. Tseng, C. Hsieh, "Prediction-based reversible data hiding," *Information Sciences*, vol. 179, no. 14, pp. 2460–2469, 2009.