sensors

MDPI

Article

Embedding Biometric Information in Interpolated Medical Images with a Reversible and Adaptive Strategy

Heng-Xiao Chi [1], Ji-Hwei Horng [2], Chin-Chen Chang [1,*] and Yung-Hui Li [3]

[1] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[2] Department of Electronic Engineering, National Quemoy University, Kinmen 89250, Taiwan
[3] AI Research Center, Hon Hai (Foxconn) Research Institute, Taipei City 114699, Taiwan
* Correspondence: alan3c@gmail.com

Abstract: How to hide messages in digital images so that messages cannot be discovered and tampered with is a compelling topic in the research area of cybersecurity. The interpolation-based reversible data hiding (RDH) scheme is especially useful for the application of medical image management. The biometric information of patients acquired by biosensors is embedded into an interpolated medical image for the purpose of authentication. The proposed scheme classifies pixel blocks into complex and smooth ones according to each block's dynamic range of pixel values. For a complex block, the minimum-neighbor (MN) interpolation followed by DIM embedding is applied, where DIM denotes the difference between the block's interpolated pixel values and the maximum pixel values. For a smooth block, the block mean (BM) interpolation is followed by a prediction error histogram (PEH) embedding and a difference expansion (DE) embedding is applied. Compared with previous methods, this adaptive strategy ensures low distortion due to embedding for smooth blocks while it provides a good payload for complex blocks. Our scheme is suitable for both medical and general images. Experimental results confirm the effectiveness of the proposed scheme. Performance comparisons with state-of-the-art schemes are also given. The peak signal to noise ratio (PSNR) of the proposed scheme is 10.32 dB higher than the relevant works in the best case.

Keywords: data hiding; interpolation; reversible data hiding; medical image

Citation: Chi, H.-X.; Horng, J.-H.; Chang, C.-C.; Li, Y.-H. Embedding Biometric Information In interpolated Medical Images with a Reversible and Adaptive Strategy. *Sensors* **2022**, *22*, 7942. https://doi.org/10.3390/s22207942

Academic Editor: Jochen Klenk

Received: 22 September 2022
Accepted: 15 October 2022
Published: 18 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Hiding information that does not want to be discovered or tampered with in digital image carriers is a hot issue in the research area of information security. According to the application purpose, it can be divided into steganography and watermarking. Typically, people use watermarking techniques for copyright protection and steganography for covert communication. Data hiding methods can also be divided into two kinds: reversible and irreversible, depending on the receiver's ability to recover the cover digital image. Reversible data hiding (RDH) [1–13] can recover the primary digital image after reading the embedded secret messages. While irreversible data hiding (IRDH) [14,15] cannot recover the primary digital image from the message-embedded confidential image completely without loss. Nevertheless, the IRDH method is usually able to embed more messages than the RDH method.

Recently, many RDH schemes based on the interpolated cover image were proposed [16–30], which can embed more secret data than traditional RDH methods while preserving a good visual quality of confidential images. The interpolation-based RDH scheme is especially useful for the application of medical image management. Since the visual quality of clinical images is directly related to the diagnosis, perceptible distortion is unacceptable. Therefore, it is more desirable to keep the original pixel value unchanged.

The cover image is usually generated by upsampling with interpolation technology in the interpolation-based RDH method. Then, the confidential message is hidden into the interpolated pixels only. In this way, after extracting the confidential data, we only need to delete the interpolated pixels from the confidential image, and we can recover the primary image perfectly.

In e-healthcare information management, if we can store and transmit the biometric information of a patient together with its corresponding medical image, an authentication procedure can be executed to ensure correct usage of the medical image. Moreover, we need to hide the information in its corresponding medical image with the least distortion to prevent misdiagnosis. The interpolation-based RDH method provides a good solution for this demand.

The key techniques for interpolation-based RDH are the interpolation method and the data embedding method. Previous work in interpolation technology includes the neighbor mean interpolation (NMI) [16–18], the interpolation by neighboring pixel (INP) [19], and the pixel repetition technology, etc. In 2021, Mandal et al. [20] published an improved interpolation technology, which gives an interpolated pixel value toward the minimum value of the block. Then, the difference between the interpolated value and the maximum value (DIM) of the block is applied to embed secret data. Since the gap is enlarged, its embedding capacity also increases. Moreover, the difference expansion (DE) [4,5] is applied to embed a second layer of confidential messages, which further increases the embedding capacity. In 2020, Geetha et al. [21] published a method that uses the last significant bit (LSB) substitution after re-interpolation, resulting in a good visual quality of the confidential images at low payloads. Many other existing RDH methods are also applicable to the interpolation-based RDH scheme.

The scheme published by Mandal et al. [20], in 2021, provides a good embedding capacity compared to other existing schemes. However, when the dynamic extent of pixel values in a block is large, the DE embedding in the second layer leads to severe distortion of the pixel value. We propose an adaptive strategy that first classifies pixel blocks into two types: smooth pixel blocks and complex pixel blocks. Then, different pixel value interpolation and data embedding strategies are designed to fit the feature of the given block. The analysis of the experimental results shows that our proposed strategy is effective both on general images and medical images.

The proposed scheme is elaborated in this paper through the following sections. Some of the related work is briefly described in Section 2. Section 3 presents the proposed adaptive RDH scheme. Section 4 not only provides its own experimental data and compares it with related works but also analyzes the experimental results. Finally, we summarize the proposed scheme in Section 5.

2. Related Works

In this section, we introduce two fundamental tools for data hiding, including difference expansion (DE) and the prediction error histogram (PEH), which are used in the related and the proposed RDH schemes. Then, an interpolation-based RDH scheme proposed by Mandal et al. is discussed, which is the main reference of our RDH scheme.

2.1. Difference Expansion (DE)

In the differential expansion (DE) [4,5] method, we process the primary image with pixel pairs as the unit of operation. For each pixel pair, the mean value and difference value are calculated first. Centered at the mean value, the difference value is evenly double-expanded outward. We can hide a bit of the confidential message in a pixel pair by switching the difference value between an adjacent pair of even and odd difference values. The formulas for DE are given by:

$$\mu = \left\lfloor \frac{s + t}{2} \right\rfloor, d = |s - t|, d' = 2 \times d + b. \quad (1)$$

$$s' = \begin{cases} \mu + \lceil d'/2 \rceil, & \text{, if } s > t, \\ \mu - \lfloor d'/2 \rfloor, & \text{otherwise.} \end{cases} \tag{2}$$

$$t' = \begin{cases} \mu - \lfloor d'/2 \rfloor & \text{, if } s > t, \\ \mu + \lceil d'/2 \rceil, & \text{otherwise.} \end{cases} \tag{3}$$

The variables (s, t) denote the cover pixel pair, μ is the average value, d is the difference, b is a binary secret bit, d' is the modified difference, and (s', t') is the marked pixel pair. When we want to extract the information, the secret bit can be read from the new pixel pair by:

$$\mu = \left\lfloor \frac{s' + t'}{2} \right\rfloor, d' = |s' - t'|, b = \mathrm{mod}(d', 2), d = \lfloor d'/2 \rfloor.$$
$$s = \begin{cases} \mu + \lceil d/2 \rceil, & \text{if } s' > t', \\ \mu - \lfloor d/2 \rfloor, & \text{otherwise.} \end{cases} \tag{4}$$
$$t = \begin{cases} \mu - \lfloor d/2 \rfloor, & \text{if } s' > t', \\ \mu + \lceil d/2 \rceil, & \text{otherwise.} \end{cases}$$

2.2. Prediction Error Histogram (PEH)

There are many PEH-based RDH techniques [8–10] and the differences between these algorithms mainly focus on two aspects, which are PEH generation and the PEH modification. For PEH generation, researchers have published various prediction schemes to improve the prediction accuracy and thus generate a more concentrated histogram; for PEH modification, researchers have also proposed many PEH shifting methods to enhance the data embedding rate of confidential images and reduce the expansion distortion. In this paper, we use a commonly applied histogram shifting method to modify PEH. After obtaining the predicted pixel value $\mathcal{D}'$, we use Equation (5) to calculate the prediction error $\mathbb{E}$ and embed the confidential message into the predicted values using the rules given in Equation (6). The corresponding process is illustrated in Figure 1a, where bins 0 and −1 are merged back to bin 0; bins 1 and 2 are merged back to 1; and secret bits are extracted during the process:

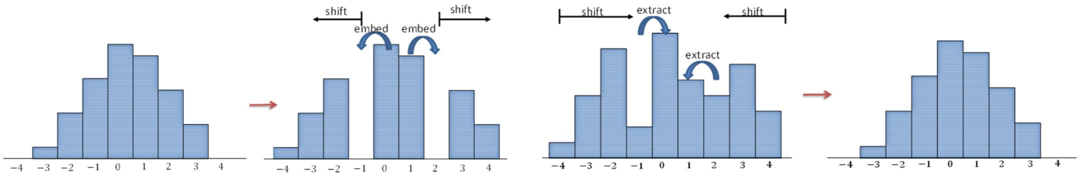
$$\mathbb{E} = \mathcal{D} - \mathcal{D}', \tag{5}$$

$$\mathcal{D}'' = \begin{cases} \mathcal{D} + b, & \text{if } \mathbb{E} = 1, \\ \mathcal{D} - b, & \text{if } \mathbb{E} = 0, \\ \mathcal{D} + 1, & \text{if } \mathbb{E} > 1, \\ \mathcal{D} - 1, & \text{if } \mathbb{E} < 0. \end{cases} \tag{6}$$

When we need to extract data, we first obtain the gap $\mathbb{E}'$ between the marked pixel-value $\mathcal{D}''$ and the predicted pixel-value $\mathcal{D}'$ by Equation (7). Then, we can read the confidential message by the rule of Equation (8) while recovering the original gap. The corresponding process is illustrated in Figure 1b, where bins 0 and 1 are selected as the target of embedding; the outside bins are shifted outward to vacate the required embedding space:

$$\mathbb{E}' = \mathcal{D}'' - \mathcal{D}', \tag{7}$$

$$\mathcal{D} = \begin{cases} \mathcal{D}'' + 1, b = 1, & \text{if } \mathbb{E}' = -1, \\ \mathcal{D}'' - 1, b = 1, & \text{if } \mathbb{E}' = 2, \\ \mathcal{D}'', b = 0, & \text{if } \mathbb{E}' = 0, \\ \mathcal{D}'', b = 0, & \text{if } \mathbb{E}' = 1, \\ \mathcal{D}'' - 1, & \text{if } \mathbb{E}' > 2, \\ \mathcal{D}'' + 1, & \text{if } \mathbb{E}' < -1. \end{cases} \tag{8}$$



(**a**) (**b**)

Figure 1. Illustration of data embedding and extraction based on histogram shifting; (**a**) Data embedding process; (**b**) Data extraction process.

2.3. Mandal et al.'s Method [20]

In the method published by Mandal et al. [20], a processing unit of the primary image is a pixel block of size 2×2. Each unit is upsampled into a block of the size 3×3. The inserted pixel values are interpolated using the minimum pixel value and its neighboring pixel values (MN). The data embedding includes two phases. In the first phase, the difference gap between the interpolated pixel value and the maximum pixel value (DIM) is exploited to embed confidential messages with an adjustable length. In the second phase, the interpolated pixels are grouped into two pairs and embedded confidential messages using DE.

An example of Mandal et al.'s RDH processing is given in Figure 2**.** As shown in Figure 2, the MN interpolation technique is first applied to calculate the three interpolated pixel values: $I(1,2) = 69$, $I(2,1) = 70$, and $I(2,2) = 69$. In the first embedding phase, DIM is used to estimate the payload of each interpolated pixel. According to the estimated payloads, secret data $s_1 = (101)_2$, $s_2 = (111)_2$, and $s_3 = (010)_2$ are embedded to obtain $I'(1,2) = 74$, $I'(2,1) = 77$, and $I'(2,2) = 71$. In the second phase, we first use $I'(1,2) = 74$ and $I'(2,2) = 71$ as a pair for DE embedding. The secret data $s_4 = 1$ is embedded to obtain $I''(1,2) = 76$ and $I''(2,2) = 69$. Then, we use $I'(2,1) = 77$ and $I''(2,2) = 69$ as a pair for DE embedding. The secret data $s_5 = 0$ is embedded to obtain $I''(2,1) = 81$ and $I'''(2,2) = 65$.

In the first phase, DIM in the processing block is exploited to embed data. The larger the difference, the larger the value available for modification, which means that more secret bits can be embedded in the gap. However, the payload would be very small when the neighboring pixel values of a cover image change smoothly. Moreover, DE embedding is applied in the second phase to further improve the ability of the program to embed information. However, the visual quality of the confidential image is greatly degraded when the pixel value difference is large.

In most medical images, the background of the image usually contains a large smooth area, which is not suitable for the application of this RDH method. Moreover, the visual quality of a medical image is a very crucial concern. Therefore, severe degradation after data embedding is not allowed.

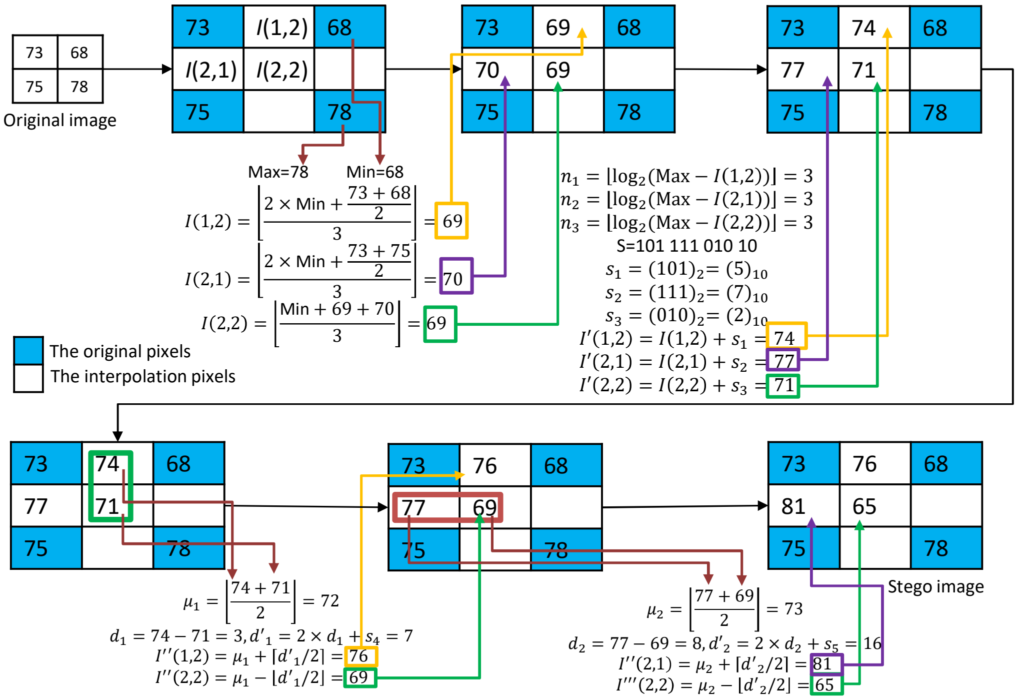

Original image
Max=78
Min=68
$I(1,2) = \left\lfloor \frac{2 \times \text{Min} + \frac{73+68}{2}}{3} \right\rfloor = 69$
$I(2,1) = \left\lfloor \frac{2 \times \text{Min} + \frac{73+75}{2}}{3} \right\rfloor = 70$
$I(2,2) = \left\lfloor \frac{\text{Min} + 69 + 70}{3} \right\rfloor = 69$
The original pixels
The interpolation pixels
$n_1 = \lfloor \log_2(\text{Max} - I(1,2)) \rfloor = 3$
$n_2 = \lfloor \log_2(\text{Max} - I(2,1)) \rfloor = 3$
$n_3 = \lfloor \log_2(\text{Max} - I(2,2)) \rfloor = 3$
S=101 111 010 10
$s_1 = (101)_2 = (5)_{10}$
$s_2 = (111)_2 = (7)_{10}$
$s_3 = (010)_2 = (2)_{10}$
$I'(1,2) = I(1,2) + s_1 = 74$
$I'(2,1) = I(2,1) + s_2 = 77$
$I'(2,2) = I(2,2) + s_3 = 71$
$\mu_1 = \left\lfloor \frac{74+71}{2} \right\rfloor = 72$
$d_1 = 74 - 71 = 3, d'_1 = 2 \times d_1 + s_4 = 7$
$I''(1,2) = \mu_1 + \lceil d'_1/2 \rceil = 76$
$I''(2,2) = \mu_1 - \lfloor d'_1/2 \rfloor = 69$
$\mu_2 = \left\lfloor \frac{77+69}{2} \right\rfloor = 73$
$d_2 = 77 - 69 = 8, d'_2 = 2 \times d_2 + s_5 = 16$
$I''(2,1) = \mu_2 + \lceil d'_2/2 \rceil = 81$
$I'''(2,2) = \mu_2 - \lfloor d'_2/2 \rfloor = 65$
Stego image

Figure 2. An example of the MN interpolation and the data embedding with the DIM and DE methods.

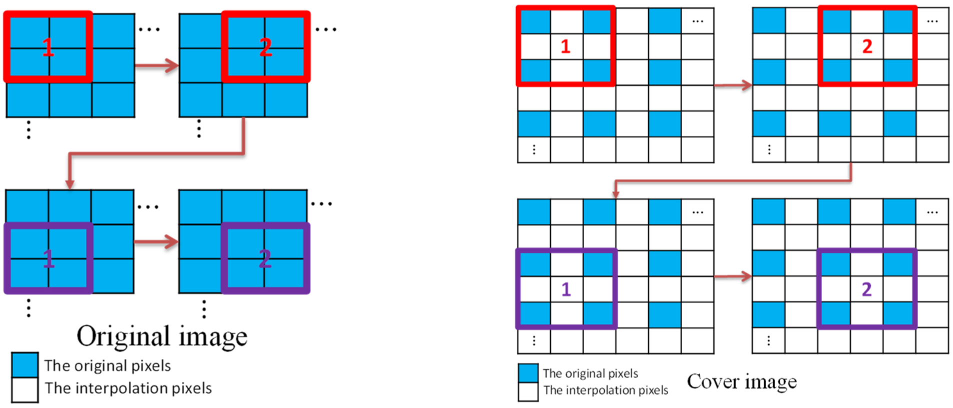
3. Proposed Scheme

A typical medical image usually contains both a large area of complex textures and a smooth background. For the image blocks with complex textures, we can take advantage of their complexity to embed multiple secret bits as in the DIM method of [20]. However, the additional DE embedding in the second phase results in a large displacement of pixel values, which results in significant visual distortion, so our scheme does not apply an additional DE embedding for such blocks. For the blocks with smooth textures, the payloads of the pixels for the DIM method are usually zero since the pixel values are very close. So, the proposed scheme adopts an alternative interpolation and embedding method to effectively embed secret bits while preserving low distortion.

As illustrated in Figure 3, we follow the interpolation-based framework proposed by Mandal et al. [20] except that the processing order of image blocks is slightly modified. The original medical image is divided into 2×2-sized blocks in an overlapped manner both row-wise and column-wise and the blocks are processed in the raster scanning order.

The dynamic range of pixel values for an image block is defined as the gray level range between the minimum and the maximum pixel values. In order to ensure that the medical images with hidden data have a good visual quality that will not lead to a misdiagnosis, the dynamic range of the pixel values within the processing block is utilized to hide the biometric information. Specifically, the interpolated pixel values modified by the data embedding should not exceed the dynamic range of the original pixel values regardless of whether they are above or below. Otherwise, false noises or contours may occur, and this may lead to misdiagnosis by the physician. However, the dynamic range of the image blocks is not the same. The blocks with a large dynamic range can be used to embed many data bits while the others are not suitable for this treatment. Therefore, we classify the pixel blocks into complex blocks and smooth blocks and process them using different methods.

The dynamic range of the pixel values in each block is calculated as shown in Figure 4. When the dynamic range of a block is larger than the given threshold, we consider the block to be a complex block; conversely, it is considered to be a smooth block. Meanwhile, the original image is upsampled to obtain a cover image with spaces to be inserted for later use. The detailed processing for image upsampling and block classification is given in Algorithm 1. According to the type of the processing block, a specified interpolation method and embedding method are applied to hide the patient's biometric information acquired by the biosensors. When a confidential image is received, the receiver first discriminates the block type and extracts the biometric data accordingly. After confirming the patient's biometric information, the confidential image is downsampled to generate the original image. The overall flowcharts of the embedding and the extraction stages are given in Figure 5.



(**a**) (**b**)

Figure 3. Image dividing, upsampling, and processing order of blocks; (**a**) Image dividing and processing order; (**b**) Processing order of upsampled image blocks.

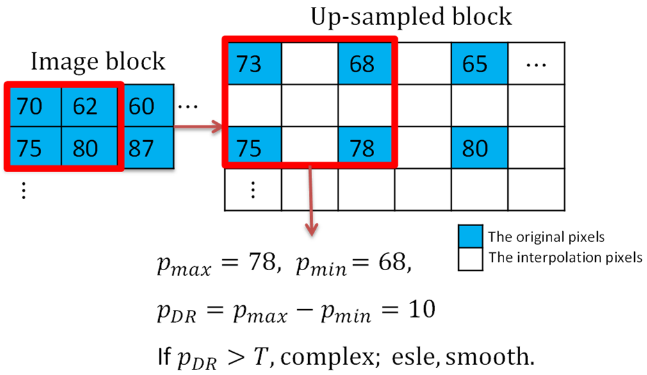
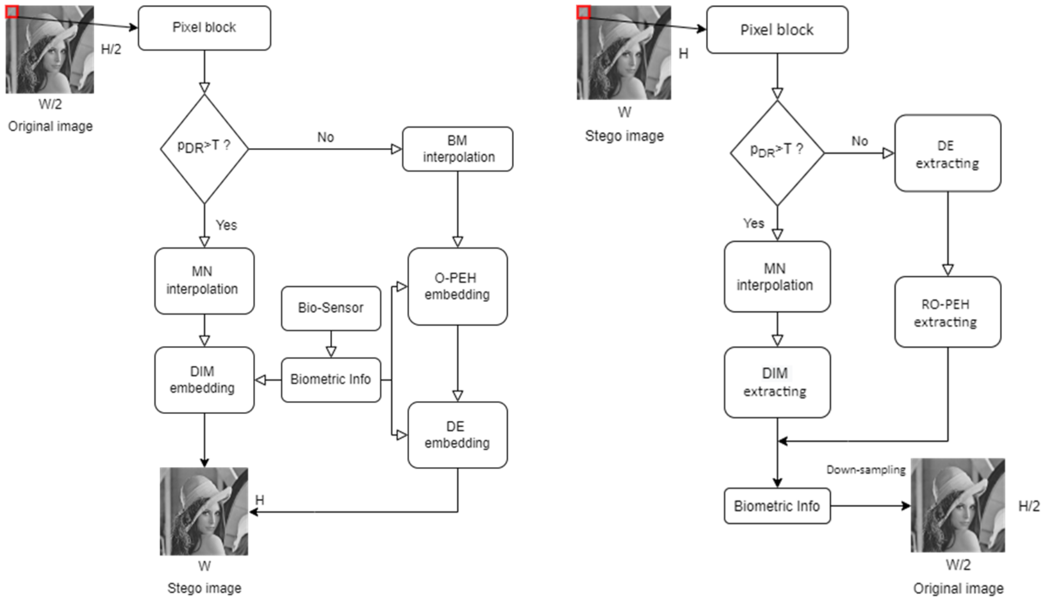

Up-sampled block
Image block
70 62 60
75 80 87
73 68 65
75 78 80
The original pixels
The interpolation pixels
$p_{max} = 78,\ p_{min} = 68,$
$p_{DR} = p_{max} - p_{min} = 10$
If $p_{DR} > T$, complex; esle, smooth.

Figure 4. An example of pixel block classification.

Algorithm 1 Image Upsampling and Block Classification

```
Input:    Original cover image $\mathbb{OI}$-sized $\mathcal{M} \times \mathcal{N}$, threshold value $\mathcal{T}$.
Output:   Vacated cover image $\mathbb{C}$-sized $(2\mathcal{M}-1) \times (2\mathcal{N}-1)$.
Step 1:   Upsample the $\mathcal{M} \times \mathcal{N}$-sized image $\mathbb{OI}$ by zero-interlacing to obtain a $(2\mathcal{M}-1) \times (2\mathcal{N}-1)$-sized image $\mathbb{CI}$, whose values are given by
          $\mathbb{C}(g,h) = \begin{cases} \mathbb{OI}((g+1)/2,(h+1)/2), g,h \in \text{odd}, \\ 0, \text{otherwise}. \end{cases}$   (9)
Step 2:   Image $\mathbb{OI}$ is divided into overlapped $2 \times 2$ pixel blocks given by
          $\mathbb{B}(s,t) = \mathbb{OI}(s:s+1, t:t+1), 1 < s < (\mathcal{M}-1), 1 < t < (\mathcal{N}-1).$   (10)
Step 3:   The dynamic range $p_{DR}$ of each block $\mathbb{B}(s,t)$ is calculated by
          $p_{max} = \max\{\mathbb{B}(s,t)\} = \max\{\mathbb{OI}(s:s+1, t:t+1)\},$
          $p_{min} = \min\{\mathbb{B}(s,t)\} = \min\{\mathbb{OI}(s:s+1, t:t+1)\},$   (11)
          $p_{DR} = p_{max} - p_{min}.$
```

If $p_{DR} > T$, $\mathbb{B}(s,t)$ is a complex block; else, it is a smooth block.



(**a**) (**b**)

Figure 5. Flowcharts of the embedding and the extraction stages; (**a**) Flowchart of the embedding stage; (**b**) Flowchart of the extraction stage.

3.1. Data Embedding Stage

After determining the type of all blocks, all the blocks $\mathbb{B}(s,t)$,$1 < s < (\mathcal{M} - 1)$, $1 < t < (\mathcal{N} - 1)$ are processed in the raster scan order. When the block $\mathbb{B}(s,t)$ is to be processed, its corresponding upsampled vacated block $\mathbb{C}(2s - 1{:}\,2s + 1{,}2t - 1{:}\,2t + 1)$ serves as the current processing unit. Depending on the block type of (s,t), two different data embedding processes for complex blocks and smooth blocks are presented as follows.

3.1.1. Data Embedding for Complex Blocks

For complex pixel blocks, the dynamic range of pixel values within the block is relatively large. The data hiding strategy is to embed EPR by filling DIM. To maximize the payload, an interpolated pixel value that is closer to the minimum pixel value of its neighbors is helpful. The MN interpolation method and DIM embedding method proposed by Mandal et al. [20] in 2020 is a successful one. We adopt MN interpolation and DIM embedding, in their first embedding phase, as the embedding method for the complex blocks in our scheme. For the convenience of interpretation, we change the dummy variables in the vacated pixel block $\mathbb{C}(2s - 1{:}\,2s + 1{,}2t - 1{:}\,2t + 1)$ into $\mathbb{C}(g{:}\,g + 2, h{:}\,h + 2)$. The data embedding process for complex blocks is given in Algorithm 2.

```
Algorithm 2 Data Embedding for Complex Blocks
Input:    Cover block $\mathbb{C}(g{:}\,g + 2, h{:}\,h + 2)$, secret data $\mathbb{S}$, maximum payload $n_{max}$.
Output:   Marked block $\hat{\mathbb{C}}(g{:}\,g + 2, h{:}\,h + 2)$.
Step 1:   Calculate the interpolated pixel values by
          $p_{min} = \min\{\mathbb{C}(g{:}\,g + 2, h{:}\,h + 2)\}$,
          $\mathbb{C}(g, h + 1) = (2 \times p_{min} + (\mathbb{C}(g, h) + \mathbb{C}(g, h + 2))/2)/3$,
          $\mathbb{C}(g + 1, h) = (2 \times p_{min} + (\mathbb{C}(g, h) + \mathbb{C}(g + 2, h))/2)/3$,      (12)
```

```
        $\mathbb{C}(g+1,h+1)=(p_{min}+\mathbb{C}(g+1,h)+\mathbb{C}(g,h+1))/3$.
Step 2: Calculate the difference values by
        $p_{max}=\max\{\mathbb{C}(g:g+2,h:h+2)\}, d_1=p_{max}-\mathbb{C}(g,h+1)$,
        $d_2=p_{max}-\mathbb{C}(g+1,h), d_3=p_{max}-\mathbb{C}(g+1,h+1)$.      (13)
Step 3: Calculate the payload for each interpolated pixel by
        $n_r=\min(\lfloor\log_2 d_r\rfloor, n_{max}), r=1,2,3$.      (14)
Step 4: Retrieve $n_1$, $n_2$, and $n_3$ secret bits from $\mathbb{S}$ and convert to decimal values $s_1$, $s_2$, and $s_3$. Then, add
        to the interpolated pixel values as
        $\hat{\mathbb{C}}(g,h+1)=\mathbb{C}(g,h+1)+s_1$,
        $\hat{\mathbb{C}}(g+1,h)=\mathbb{C}(g+1,h)+s_2$,      (15)
        $\hat{\mathbb{C}}(g+1,h+1)=\mathbb{C}(g+1,h+1)+s_3$.
```

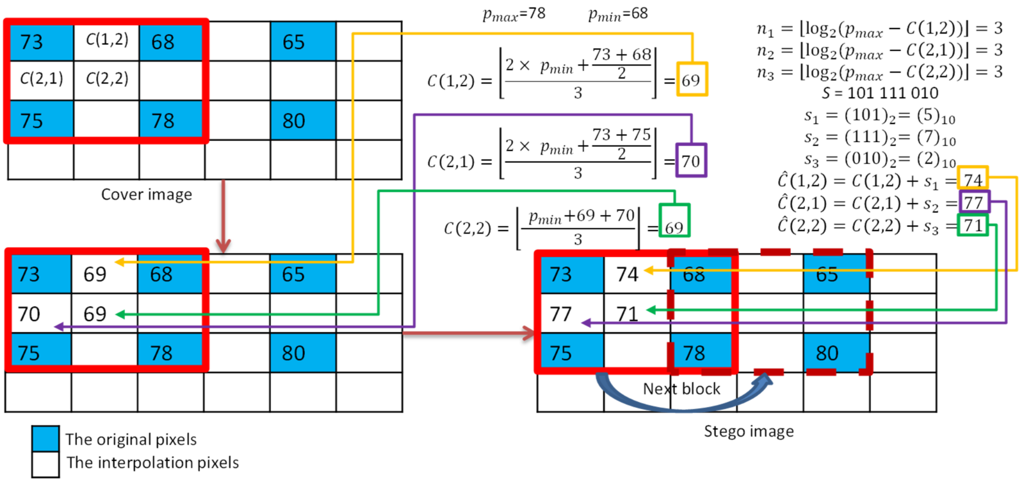
As shown in Figure 6, we give an example to illustrate the interpolation and the DIM data hiding strategy of our scheme for a complex block. Three interpolated pixel values $\mathbb{C}(1,2)=69$, $\mathbb{C}(2,1)=70$, and $\mathbb{C}(2,2)=69$ are calculated using MN interpolation. Then, the payloads for each interpolated pixel are calculated. According to the payloads, the data $s_1=(101)_2=5$, $s_2=(111)_2=7$, and $s_3=(010)_2=2$ are retrieved from the binary secret stream. Finally, the marked pixel values $\hat{\mathbb{C}}(1,2)=74$, $\hat{\mathbb{C}}(2,1)=77$ and $\hat{\mathbb{C}}(2,2)=71$ can be calculated.

3.1.2. Data Embedding for Smooth Blocks

For smooth pixel blocks, the dynamic range of pixel values in the block is relatively small. This characteristic provides a very good basis for PEH and DE embedding. Therefore, we design a completely different strategy for pixel value interpolation and data embedding of smooth blocks.

The proposed scheme uses the block mean (BM) as the interpolated pixel values for smooth blocks. Then, the data embedding procedure includes two layers, an ordered PEH (O-PEH) embedding and a DE embedding. For the O-PEH embedding, three interpolated pixels are embedded with secret data in a predefined order. The resulting pixel values are further grouped into two pairs and embedded with secret data using DE.

Since the interpolated pixel values are mutually equal, they are always embeddable, using O-PEH, in the first embedding layer. The resulting pixel values deviate by no more than 2; therefore, the expansion range of DE embedding is also very small. The overall embedding strategy can ensure a good payload and small distortion of the pixel values. The data embedding procedure for smooth blocks is summarized in Algorithm 3.



73
C(1,2)
68
65
C(2,1)
C(2,2)
75
78
80
Cover image
73
69
68
65
70
69
75
78
80
p_{max}=78
p_{min}=68
$C(1,2)=\left\lfloor\frac{2\times p_{min}+\frac{73+68}{2}}{3}\right\rfloor=\boxed{69}$
$C(2,1)=\left\lfloor\frac{2\times p_{min}+\frac{73+75}{2}}{3}\right\rfloor=\boxed{70}$
$C(2,2)=\left\lfloor\frac{p_{min}+69+70}{3}\right\rfloor=\boxed{69}$
$n_1=\lfloor\log_2(p_{max}-C(1,2))\rfloor=3$
$n_2=\lfloor\log_2(p_{max}-C(2,1))\rfloor=3$
$n_3=\lfloor\log_2(p_{max}-C(2,2))\rfloor=3$
S = 101 111 010
$s_1=(101)_2=(5)_{10}$
$s_2=(111)_2=(7)_{10}$
$s_3=(010)_2=(2)_{10}$
$\hat{C}(1,2)=C(1,2)+s_1=\boxed{74}$
$\hat{C}(2,1)=C(2,1)+s_2=\boxed{77}$
$\hat{C}(2,2)=C(2,2)+s_3=\boxed{71}$
73
74
68
65
77
71
75
78
80
Next block
Stego image
The original pixels
The interpolation pixels

Figure 6. Example of interpolation and data embedding for complex blocks.

Algorithm 3 Data Embedding for Smooth Blocks

```
Input:    Cover block $\mathbb{C}(g:g+2, h:h+2)$, secret data $\mathbb{S}$.
Output:   Stego block $\hat{\mathbb{C}}(g:g+2, h:h+2)$.
Step 1:   Execute the BM interpolation by
              $BM = \frac{\mathbb{C}(g,h)+\mathbb{C}(g+2,h)+\mathbb{C}(g,h+2)+\mathbb{C}(g+2,h+2)}{4}$
              $\mathbb{C}(g,h+1)=\mathbb{C}(g+1,h)=\mathbb{C}(g+1,h+1)=BM.$                    (16)
Step 2:   Embed layer 1:
          Calculate PEs of the interpolated pixels in the order given by
              $PE_1=\mathbb{C}(g+1,h+1)-\mathbb{C}(g,h+1),$
              $PE_2=\mathbb{C}(g,h+1)-\mathbb{C}'(g+1,h+1),$                                   (17)
              $PE_3=\mathbb{C}(g+1,h)-\mathbb{C}'(g+1,h+1).$
          Embed 1-bit of secret data $b$ from $S$ to the leading pixel in (16) by
              $\mathbb{C}'=\begin{cases}\mathbb{C}+b, & \text{if } PE=1\\ \mathbb{C}-b, & \text{if } PE=0\\ \mathbb{C}+1, & \text{if } PE>1\\ \mathbb{C}-1, & \text{if } PE<0\end{cases}$   (18)
Step 3:   Embed layer 2:
          Apply DE to embed 1-bit of secret data from $S$ according to the following processes
              $\{\mathbb{C}'(g,h+1),\mathbb{C}'(g+1,h+1)\}\overset{DE}{\rightarrow}\{\mathbb{C}''(g,h+1),\mathbb{C}''(g+1,h+1)\},$
              $\{\mathbb{C}'(g+1,h),\mathbb{C}''(g+1,h+1)\}\overset{DE}{\rightarrow}\{\mathbb{C}''(g+1,h),\mathbb{C}'''(g+1,h+1)\}.$   (19)
Step 4:   Record $\mathbb{C}''(g,h+1),\mathbb{C}''(g+1,h)$, and $\mathbb{C}'''(g+1,h+1)$ as $\hat{\mathbb{C}}(g,h+1),\hat{\mathbb{C}}(g+1,h)$ and $\hat{\mathbb{C}}(g+1,h+1)$, respectively.
```

As shown in Figure 7, we give an example to illustrate the BM interpolation and two-layer embedding strategy for smooth pixel blocks. The seed pixels are 73, 68, 75, and 78; and the secret data stream is assumed to be '10,110'. We first use the block mean as the interpolated pixel values $\mathbb{C}(1{,}2) = \mathbb{C}(2{,}1) = \mathbb{C}(2{,}2) = 74$. Then, the proposed O-PEH is applied first. The secret data $b_1 = 1$ is embedded to $\mathbb{C}(2{,}2)$ and it results in $\mathbb{C}'(2{,}2) = 73$. The next two secret bits $b_2 = 0$ and $b_3 = 1$ are then embedded to $\mathbb{C}(1{,}2)$ and $\mathbb{C}(2{,}1)$, and the resulting pixel values are $\mathbb{C}'(1{,}2) = 74$ and $\mathbb{C}'(2{,}1) = 75$. In the second embedding layer, we first use $\mathbb{C}'(1{,}2)$ and $\mathbb{C}'(2{,}2)$ as a group for DE embedding and embed secret data $b_4 = 1$, and $\mathbb{C}''(1{,}2) = 75$ and $\mathbb{C}''(2{,}2) = 72$ are obtained. Finally, $\mathbb{C}'(2{,}1)$ and $\mathbb{C}''(2{,}2)$ are used as a group for DE embedding and to embed secret data $b_5 = 0$, and $\mathbb{C}''(2{,}1) = 76$ and $\mathbb{C}'''(2{,}2) = 70$ are obtained.

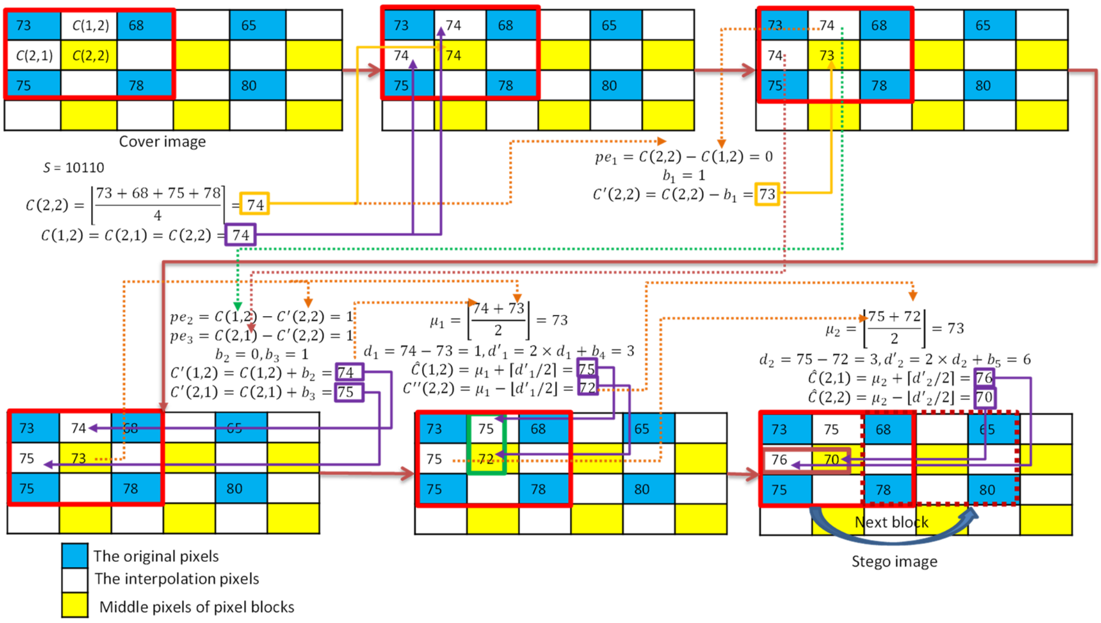

Cover image
S = 10110
C(2,2) = ⌈(73 + 68 + 75 + 78)/4⌉ = 74
C(1,2) = C(2,1) = C(2,2) = 74
pe₁ = C(2,2) − C(1,2) = 0
b₁ = 1
C′(2,2) = C(2,2) − b₁ = 73
pe₂ = C(1,2) − C′(2,2) = 1
pe₃ = C(2,1) − C′(2,2) = 1
b₂ = 0, b₃ = 1
C′(1,2) = C(1,2) + b₂ = 74
C′(2,1) = C(2,1) + b₃ = 75
μ₁ = ⌊(74 + 73)/2⌋ = 73
d₁ = 74 − 73 = 1, d′₁ = 2 × d₁ + b₄ = 3
Ĉ(1,2) = μ₁ + ⌈d′₁/2⌉ = 75
C″(2,2) = μ₁ − ⌊d′₁/2⌋ = 72
μ₂ = ⌊(75 + 72)/2⌋ = 73
d₂ = 75 − 72 = 3, d′₂ = 2 × d₂ + b₅ = 6
Ĉ(2,1) = μ₂ + ⌈d′₂/2⌉ = 76
Ĉ(2,2) = μ₂ − ⌊d′₂/2⌋ = 70
Next block
Stego image
The original pixels
The interpolation pixels
Middle pixels of pixel blocks

Figure 7. Example of interpolation and data embedding for smooth blocks.

3.2. Data Extraction Stage

After people receive a stego medical image, the stego medical image is first divided into pixel blocks sized 3×3 that overlap between rows and columns with a stride of 2 to ensure that the processing units are exactly the same as the embedding. Then, the dynamic range of pixel values for each block is calculated to classify the blocks into complex and smooth types. The data extraction algorithms for complex and smooth blocks are described in the following subsections. After extracting the secret data, the confidential image is downsampled to obtain the original medical image.

3.2.1. Data Extraction for Complex Pixel Blocks

For a complex pixel block, we first calculate the interpolated pixel values using the seed pixel values so that we can obtain the gap between each original interpolated pixel value and the maximum value. The payload for each interpolated pixel is therefore obtained. Then, we obtain our secret data by converting the difference between the confidential pixel value and the maximum value back to the binary secret bits with the length determined by its payload. The extraction procedure of complex blocks is summarized in Algorithm 4.

An example of data extraction for complex blocks is shown in Figure 8. The original interpolated pixel values $\mathbb{C}(1,2) = 69$, $\mathbb{C}(2,1) = 70$, and $\mathbb{C}(2,2) = 69$ are calculated first. Then, the difference values with respect to $p_{max} = 78$ are calculated to determine the payloads. Finally, the difference between each stego pixel value and original interpolated value is calculated and converted to binary bits according to its payload as shown in the figure.

Algorithm 4 Data Extraction for Complex Blocks

```
Input:    Stego block $\hat{\mathbb{C}}(g: g+2, h: h+2)$, maximum payload $n_{max}$.
Output:   Secret data $\mathbb{S}$.
Step 1:   Calculate the original interpolated pixel values by
          $\mathbb{C}(g, h+1) = (2 \times p_{min} + (\hat{\mathbb{C}}(g, h) + \hat{\mathbb{C}}(g, h+2))/2)/3,$
          $\mathbb{C}(g+1, h) = (2 \times p_{min} + (\hat{\mathbb{C}}(g, h) + \hat{\mathbb{C}}(g+2, h))/2)/3,$          (20)
          $\mathbb{C}(g+1, h+1) = (p_{min} + \hat{\mathbb{C}}(g+1, h) + \hat{\mathbb{C}}(g, h+1))/3.$
Step 2:   Calculate the difference values by
Step 3:    $d_1 = p_{max} - \mathbb{C}(g, h+1)$,
           $d_2 = p_{max} - \mathbb{C}(g+1, h)$,        (21)
           $d_3 = p_{max} - \mathbb{C}(g+1, h+1)$.
           Calculate the payload of each interpolated pixel by
           $n_r = \min(\lfloor \log_2 d_r \rfloor, n_{max}), r = 1,2,3.$        (22)
Step 4:    Calculate the difference values and restore secret bits according to $n_r$.
           $s_1 = \hat{\mathbb{C}}(g, h+1) - \mathbb{C}(g, h+1)$,
           $s_2 = \hat{\mathbb{C}}(g+1, h) - \mathbb{C}(g+1, h)$,        (23)
           $s_3 = \hat{\mathbb{C}}(g+1, h+1) - \mathbb{C}(g+1, h+1)$.
```

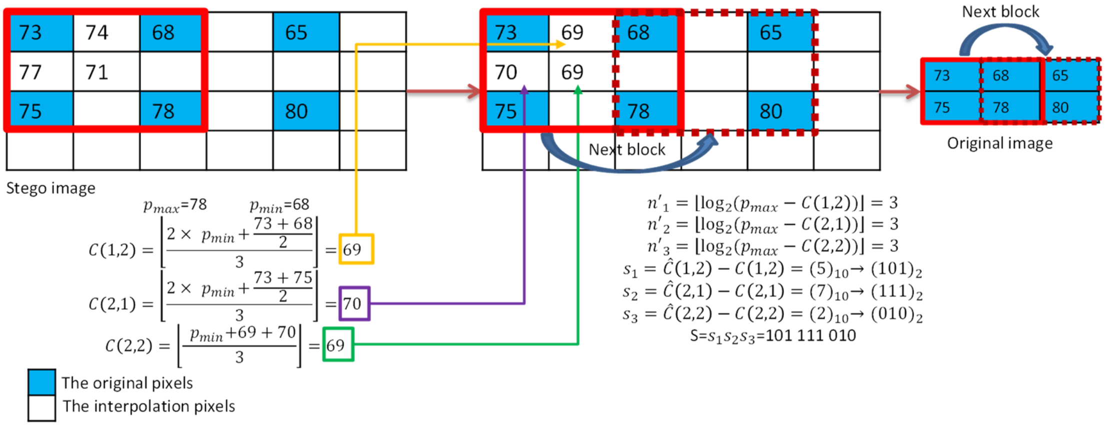

73
74
68
65
77
71
75
78
80
Stego image
73
69
68
65
70
69
75
78
80
Next block
Next block
73
68
65
75
78
80
Original image
p_{max}=78
p_{min}=68
$C(1,2) = \left\lfloor \frac{2 \times p_{min} + \frac{73+68}{2}}{3} \right\rfloor = 69$
$C(2,1) = \left\lfloor \frac{2 \times p_{min} + \frac{73+75}{2}}{3} \right\rfloor = 70$
$C(2,2) = \left\lfloor \frac{p_{min} + 69 + 70}{3} \right\rfloor = 69$
$n'_1 = \lfloor \log_2(p_{max} - C(1,2)) \rfloor = 3$
$n'_2 = \lfloor \log_2(p_{max} - C(2,1)) \rfloor = 3$
$n'_3 = \lfloor \log_2(p_{max} - C(2,2)) \rfloor = 3$
$s_1 = \hat{C}(1,2) - C(1,2) = (5)_{10} \rightarrow (101)_2$
$s_2 = \hat{C}(2,1) - C(2,1) = (7)_{10} \rightarrow (111)_2$
$s_3 = \hat{C}(2,2) - C(2,2) = (2)_{10} \rightarrow (010)_2$
S=$s_1 s_2 s_3$=101 111 010
The original pixels
The interpolation pixels

Figure 8. Example of data extraction for complex blocks.

3.2.2. Data Extraction for Smooth Pixel Blocks

For a smooth pixel block, the data extraction operation is executed according to the reverse embedding order of the pixel block. We first restore the secret bits hidden in the second layer. In this layer, the second pair of pixels is processed and then the first pair. Finally, the secret bits hidden in the first layer are also extracted in the reverse order of PEH (RO-PEH). Algorithm 5 summarizes the data extraction process of the smooth block.

Algorithm 5 Data Extraction for Smooth Blocks

```
Input:     Stego block $\hat{\mathbb{C}}(g: g+2, h: h+2)$.
Output:    Secret data $\mathbb{S}$.
           Extract layer 2:
Step 1:    Apply RDE, (4), to extract secret data according to the following steps
           $\{\hat{\mathbb{C}}(g+1, h), \hat{\mathbb{C}}(g+1, h+1)\} \xrightarrow{RDE} \{\mathbb{C}'(g+1, h), \mathbb{C}'(g+1, h+1)\}$,
           $\{\hat{\mathbb{C}}(g, h+1), \mathbb{C}'(g+1, h+1)\} \xrightarrow{RDE} \{\mathbb{C}'(g, h+1), \mathbb{C}''(g+1, h+1)\}$.        (24)
           *Extract layer 1*:
Step 2:    Apply RPEH, Equations (7) and (8), to extract secret data according to the following steps
           $\{\mathbb{C}'(g+1, h), \mathbb{C}''(g+1, h+1)\} \xrightarrow{RPEH} \{\mathbb{C}''(g+1, h), \mathbb{C}''(g+1, h+1)\}$,
           $\{\mathbb{C}'(g, h+1), \mathbb{C}''(g+1, h+1)\} \xrightarrow{RPEH} \{\mathbb{C}''(g, h+1), \mathbb{C}''(g+1, h+1)\}$,        (25)
           $\{\mathbb{C}''(g+1, h+1), \mathbb{C}''(g, h+1)\} \xrightarrow{RPEH} \{\mathbb{C}'''(g+1, h+1), \mathbb{C}''(g, h+1)\}$.
```

An example of data extraction for smooth blocks is shown in Figure 9. In the extraction of the second layer, the stego pixel pair $\{\hat{\mathbb{C}}(2,1) = 76, \hat{\mathbb{C}}(2,2) = 70\}$ is applied to extract $b_5 = 0$ and recover to $\{\mathbb{C}'^{(2,1)} = 75, \mathbb{C}'(2,2) = 72\}$. Then, the pixel pair $\{\hat{\mathbb{C}}(1,2) = 75, \mathbb{C}'(2,2) = 72\}$ is applied to extract $b_4 = 1$ and recover to $\{\mathbb{C}'(1,2) = 74, \mathbb{C}''(2,2) = 73\}$. The extraction of the first layer is processed as follows. The prediction errors of pixel $\mathbb{C}'(2,1) = 75$ and $\mathbb{C}'(1,2) = 74$ with respect to $\mathbb{C}''(2,2) = 73$ are calculated to extract $b_3 = 1, b_2 = 0$ and recover to $\mathbb{C}''(2,1) = 74, \mathbb{C}''(1,2) = 74$. Finally, the prediction error of pixel $\mathbb{C}''(2,2) = 73$ with respect to $\mathbb{C}''(1,2) = 74$ is calculated to extract $b_1 = 1$ and recover to $\mathbb{C}''(2,2) = 74$.

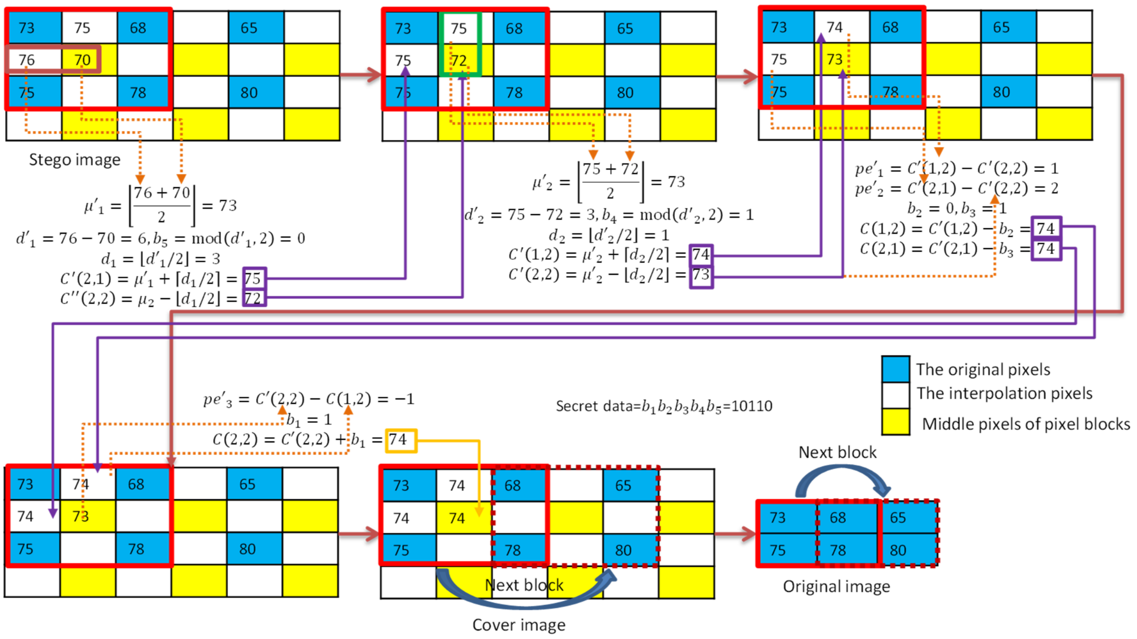

Stego image
$\mu'_1 = \left\lfloor \frac{76+70}{2} \right\rfloor = 73$
$d'_1 = 76 - 70 = 6, b_5 = \mathrm{mod}(d'_1, 2) = 0$
$d_1 = \lfloor d'_1/2 \rfloor = 3$
$C'(2,1) = \mu'_1 + \lceil d_1/2 \rceil = 75$
$C''(2,2) = \mu_2 - \lfloor d_1/2 \rfloor = 72$
$\mu'_2 = \left\lfloor \frac{75+72}{2} \right\rfloor = 73$
$d'_2 = 75 - 72 = 3, b_4 = \mathrm{mod}(d'_2, 2) = 1$
$d_2 = \lfloor d'_2/2 \rfloor = 1$
$C'(1,2) = \mu'_2 + \lceil d_2/2 \rceil = 74$
$C'(2,2) = \mu'_2 - \lfloor d_2/2 \rfloor = 73$
$pe'_1 = C'(1,2) - C'(2,2) = 1$
$pe'_2 = C'(2,1) - C'(2,2) = 2$
$b_2 = 0, b_3 = 1$
$C(1,2) = C'(1,2) - b_2 = 74$
$C(2,1) = C'(2,1) - b_3 = 74$
$pe'_3 = C'(2,2) - C(1,2) = -1$
$b_1 = 1$
$C(2,2) = C'(2,2) + b_1 = 74$
Secret data=$b_1b_2b_3b_4b_5$=10110
The original pixels
The interpolation pixels
Middle pixels of pixel blocks
Next block
Next block
Cover image
Original image

Figure 9. Example of data extraction for smooth blocks.

3.3. Overflow and Underflow

In the process of embedding secret information, extreme pixel values may cause overflow or underflow problems. In the embedding process of a complex block, pixel values are always modified within the dynamic range of the seed pixel values. Therefore, there is no overflow/underflow problem.

The smooth pixel blocks of the block mean valued 0 and 255 are left unembedded to prevent overflow/underflow of the first-layer embedding while the smooth pixel blocks of the block mean valued within the ranges 1 to 3 and 253 to 254 are only processed with the first-layer embedding to prevent the overflow/underflow of the second-layer embedding.

4. Experimental Results and Discussion

Our proposed adaptive RDH scheme was implemented on the Windows PC operating system using MATLAB version R2017a. We first verified the feasibility of our proposed adaptive RDH scheme with nine standard grayscale test images sized 512 × 512, and then tested it on the six medical images shown in Figure 10. We regarded the binary secret data $\mathbb{S}$ produced by a random number generator as the EPR to be embedded in the image. The threshold for block type classification was set to 5. The performance of the adaptive RDH scheme was evaluated with the peak signal to noise ratio (PSNR), the structural similarity (SSIM), and the embedding capability (EC). PSNR is a metric used to measure the visual quality of confidential images. Here, we adopted the PSNR definition in [2] as follows:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{H \times W \times (255)^2}{\sum_{i=1}^{H}\sum_{j=1}^{W}(G_{ij} - \hat{G}_{ij})^2}\right), \tag{26}$$

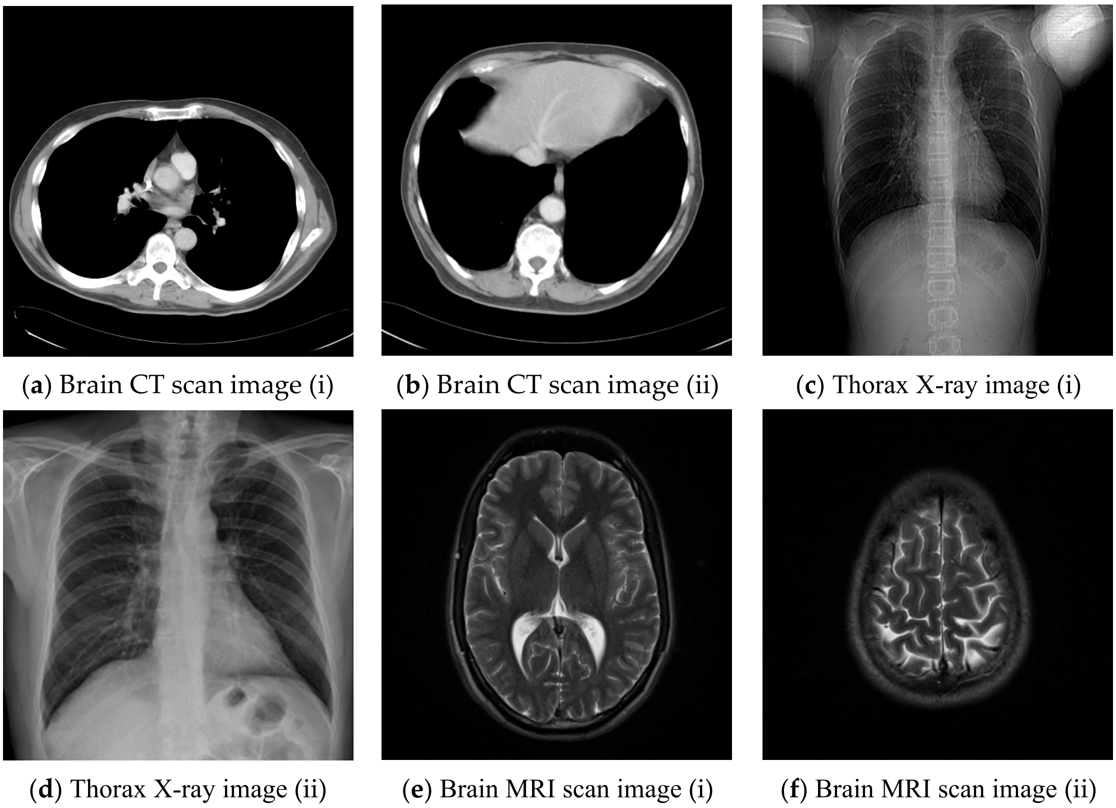
where $H \times W$ is the size of the cover image, and G_{ij} and $\hat{G}_{ij}$ denote the pixel values of the cover image and the stego-image, respectively. To know whether the confidential image is close to a natural image or not, we first downsample a test image to obtain the small image. Then, it is processed by the proposed scheme to obtain a confidential image. Finally, the PSNR value of the confidential image is measured according to the given test image. Through the experiments, if the PSNR value is larger than 30 dB, it means that the distortion caused by information embedding is not detectable by the human eye. Of course, the higher the PSNR value, the less distortion due to hidden data. By minimizing the alteration of the pixel values, confidential images with a high PSNR value can be obtained.

SSIM is also a metric used to evaluate the similarity between the original image and the stego-image, as shown in Equation (27):

$$\mathrm{SSIM} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{({\mu_x}^2 + {\mu_y}^2 + c_1)({\sigma_x}^2 + {\sigma_y}^2 + c_2)} \tag{27}$$

where μ_x and μ_y denote the mean of images x and y, respectively; σ_x, σ_y, σ_{xy} denote the standard deviation and covariance of images x and y, respectively; and c_1 and c_2 are two constants to avoid division by zero. The SSIM metric takes into account both image luminance, contrast, and structure information, which is closer to the benchmark of the human visual system; the higher the SSIM value, the more similar the original image and the stego image are.

EC, measured in bits per pixel, is defined as the total amount of payload in bits divided by the total amount of pixels in the interpolated cover image.


(**a**) Brain CT scan image (i) (**b**) Brain CT scan image (ii) (**c**) Thorax X-ray image (i)

(**d**) Thorax X-ray image (ii) (**e**) Brain MRI scan image (i) (**f**) Brain MRI scan image (ii)

Figure 10. Six medical test images; (**a**) Brain CT scan image (i); (**b**) Brain CT scan image (ii); (**c**) Thorax X-ray image (i); (**d**) Thorax X-ray image (ii); (**e**) Brain MRI scan image (i); (**f**) Brain MRI scan image (ii).

4.1. Experimental Results for Standard Grayscale Images

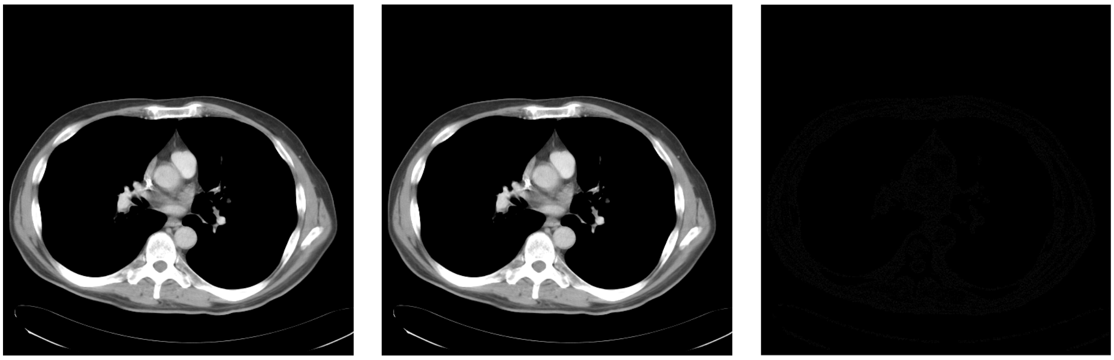
Our proposed adaptive RDH scheme was compared with the interpolation-based RDH schemes, proposed by Mandal et al. [20] and Geetha et al. [21], using the standard grayscale images. The values of EC, PSNR, and SSIM for the compared schemes are listed in Table 1. Since the scheme proposed by Geetha et al. [21] uses LSB substitution, its embedding capability is a fixed value and the features of the pixel blocks are not taken into consideration. In comparison with [21], our scheme provides a better EC with a comparable SSIM. Although the PSNR value of our scheme is lower than that of [21], the distortion is imperceptible when its value is higher than 30 dB. Mandal et al.'s scheme [20] provides the highest EC among the three schemes; however, its PSNR and SSIM are the worst. Our proposed adaptive RDH scheme has a good balance between PSNR and EC by taking the block feature into consideration. Numerically, the PSNR values obtained by our proposed adaptive RDH scheme are all greater than 30 dB, which means that the difference between the confidential image obtained by the proposed scheme and the original test image is relatively small, and the confidential image looks like a natural one.

Table 1. Comparison of PSNR, SSIM and EC with related works.

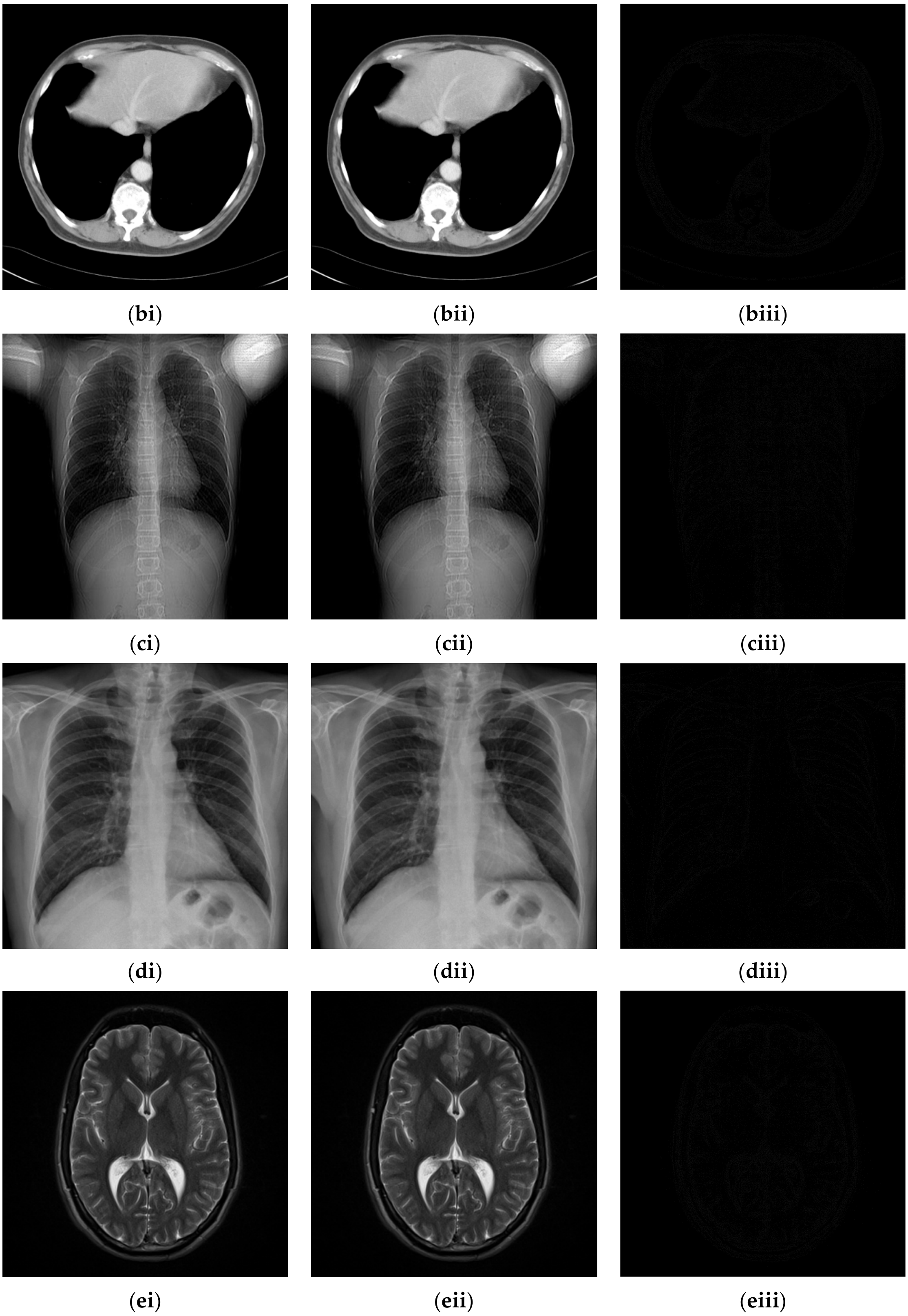
Images	**Proposed Scheme**			**[20]**			**[21]**		
	EC	**PSNR**	**SSIM**	**EC**	**PSNR**	**SSIM**	**EC**	**PSNR**	**SSIM**
Airplane	1.93	35.30	0.95	2.17	33.23	0.92	1.50	39.46	0.94
Baboon	2.73	31.57	0.94	3.19	29.40	0.86	1.50	39.44	0.97
Boat	2.30	33.68	0.94	2.70	31.65	0.86	1.50	39.43	0.95
Couple	2.33	33.49	0.94	2.56	32.05	0.87	1.50	39.41	0.95
Elaine	2.29	34.06	0.92	2.50	33.10	0.83	1.50	39.45	0.94
Goldhill	2.32	33.56	0.93	2.72	31.62	0.84	1.50	39.43	0.95
Lena	2.05	35.00	0.94	2.39	33.03	0.88	1.50	39.44	0.94
Man	2.35	33.22	0.92	2.73	31.15	0.85	1.50	39.49	0.95
Peppers	2.06	35.26	0.94	2.42	33.13	0.88	1.50	39.42	0.94

4.2. Experimental Results for Medical Images

Since the interpolation-based RDH schemes are designed for the special purpose of embedding patient information into medical images, the performance of our proposed adaptive RDH scheme on medical images is the main concern. Figure 11 shows the original image and its corresponding confidential image with 2 bpp secret data embedded, and images showing the difference between the original medical images and their confidential images. The difference images (aiii–fiii) in Figure 11 show almost nothing, which indicates that the proposed scheme does not look much different from the original image after hiding and will not draw the eavesdropper's attention.



(**ai**) (**aii**) (**aiii**)



(**bi**) (**bii**) (**biii**)

(**ci**) (**cii**) (**ciii**)

(**di**) (**dii**) (**diii**)

(**ei**) (**eii**) (**eiii**)

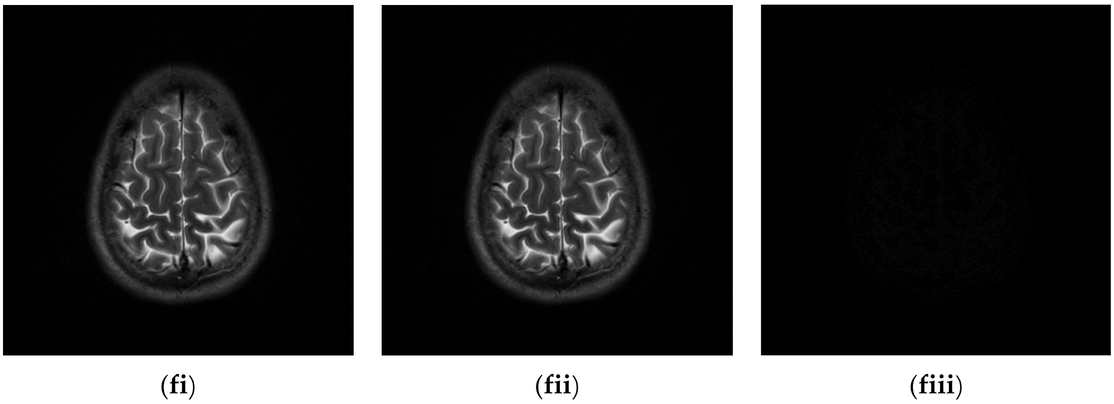

(**fi**) (**fii**) (**fiii**)

Figure 11. The original images, the confidential images, and the different images; (**ai,bi,ci,di,ei**,**fi**) The cover images; (**aii,bii,cii,dii,eii**,**fii**) The confidential images; (**aiii,biii,ciii,diii,eiii**,**fiii**) The difference between the cover images and the confidential images.

Table 2 shows the PSNR values of our proposed adaptive RDH scheme for the medical images in Figure 10, where the embedding rate ranges from 0.25 to 2 bpp. From the table, we can clearly see that PSNR of the proposed scheme reaches 53.3405 dB in a low embedding rate, which indicates that our proposed adaptive RDH scheme does not cause significant distortion to the images. In a high embedding rate, PSNR of our proposed adaptive RDH scheme is still more than 36 dB.

Table 2. PSNR for different EC values.

EC **Images**	**0.25 bpp**	**0.50 bpp**	**0.75 bpp**	**1.00 bpp**	**1.25 bpp**	**1.50 bpp**	**1.75 bpp**	**2.00 bpp**
CT (i)	53.3405	44.0953	41.1997	39.4263	38.2624	38.2767	38.2376	38.2550
CT (ii)	45.6208	42.5201	40.9425	39.7861	38.9088	38.1414	37.6155	37.5999
X-ray (i)	47.0919	43.2664	40.9943	39.4476	38.1728	37.3442	37.2253	37.2143
X-ray (ii)	52.8555	45.1903	42.0803	39.9389	39.0696	39.1002	39.1011	39.0826
MRI (i)	46.0622	42.4452	40.6458	39.3618	38.0475	37.7737	37.7601	37.7896
MRI (ii)	43.7435	41.0151	39.3316	38.1186	37.2286	36.7140	36.4273	36.3968

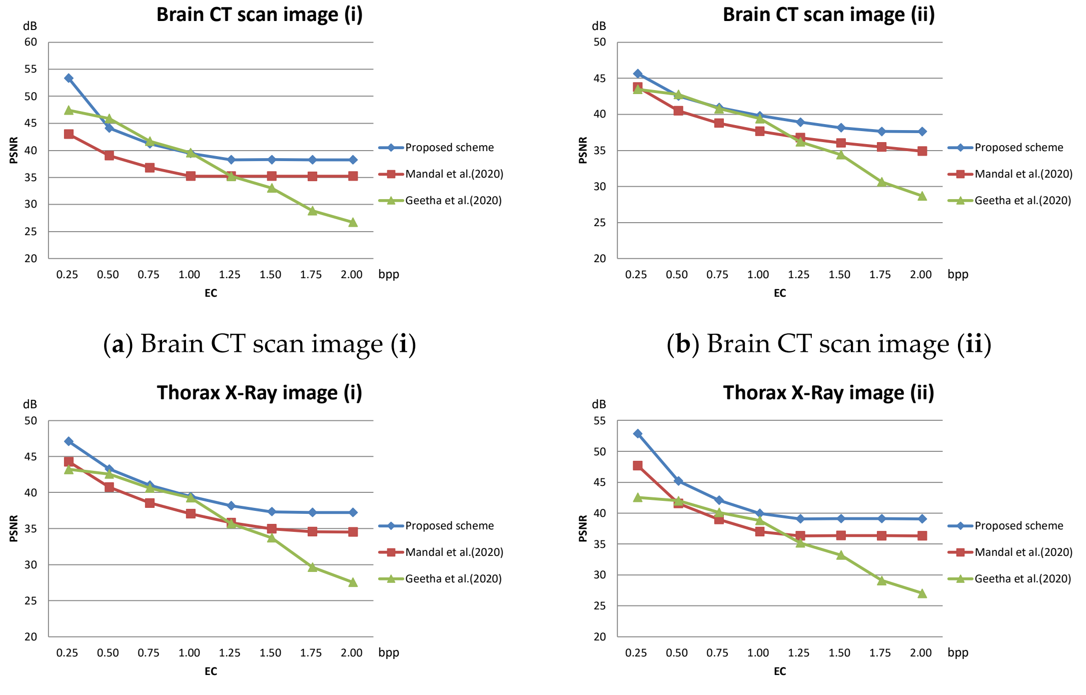
A comparison of the proposed scheme with related works is given in Table 3, where p and q represent the number of bits that can be embedded in edge pixels and non-edge pixels in [27], respectively, and k is the number of bits per embedding in [30]. From the table, we can find that our proposed adaptive RDH scheme significantly outperforms most of the related works in terms of both EC and PSNR, which means that our scheme can hide more biometric information while preserving a better visual quality than the others. For [19,26,29], our scheme wins over these schemes both in terms of EC and PSNR. In [27], when $p = 4, q = 3$, EC of [27] is similar to that of our proposed adaptive RDH scheme, but PSNR of [27] is worse than that of our proposed adaptive RDH scheme. When $p = 4, q = 4$, EC of [27] can reach 3 bpp and PSNR of the image is obviously much worse. While the proposed scheme is slightly inferior to [27] in terms of EC, we are able to maintain PSNR of the image at a relatively good level. In [30], when $k = 2$, the scheme loses to the proposed scheme in both EC and PSNR. When $k = 3$, although the proposed scheme is slightly inferior to [30] in terms of EC, the higher value of PSNR of our steganographic image makes the image carrying information less detectable during transmission.

Table 3. Comparison of PSNR and EC with related works.

Images	**Proposed**		**[19]**		**[26]**		**[27] (*p* = 4, q = 3)**		**[27] (*p* = 4, q = 4)**		**[29]**		**[30] (k = 2)**		**[30] (k = 3)**	
	EC	**PSNR**	**EC**	**PSNR**	**EC**	**PSNR**	**EC**	**PSNR**	**EC**	**PSNR**	**EC**	**PSNR**	**EC**	**PSNR**	**EC**	**PSNR**
Airplane	1.93	35.30	1.34	22.91	0.87	27.37	2.33	33.06	3.00	30.85	1.5	26.47	1.48	32.08	2.95	31.92
Baboon	2.73	31.57	2.27	20.49	1.75	21.22	2.41	29.60	3.00	28.58	1.5	25.55	\	\	\	\
Lena	2.05	35.00	1.32	22.32	0.86	30.24	2.33	34.40	3.00	31.56	1.5	26.85	1.48	33.45	2.95	33.23
Peppers	2.06	35.26	1.29	22.45	0.85	28.04	2.32	33.66	3.00	31.17	\	\	1.48	31.00	2.20	30.88

The PSNR value with respect to the payload of three related schemes for the test images given in Figure 10 are plotted in Figure 12. As shown in the figures, our scheme maintains a superior visual quality under an equal payload. When the payload is larger than 1.5 bpp, the PSNR value of Geetha et al.'s scheme [21] degrades significantly.

The method published by Mandal et al. [20] treats all pixel blocks the same. When executing DE embedding in its second data hiding phase, the large pixel value differences cause the pixel values to be drastically modified, which significantly reduces the PSNR value of the stego image. In our scheme, the pixel blocks are classified into smooth and complex ones in advance. During data hiding, DE embedding of complex blocks is omitted, which effectively improves the PSNR value. For the smooth blocks, the block mean interpolation ensures a controllable difference value between interpolated pixels and DE can be safely executed without leading to significant distortion. The experimental data confirms the effectiveness of the proposed strategy. With this strategy, our scheme is not only able to achieve good visual quality scores on general grayscale images but is also able to embed more secret messages on medical images. The proposed scheme provides a high PSNR value at a low hiding volume. As the embedded volume increases, PSNR of the proposed scheme does not decrease drastically as in [21] but decreases smoothly.

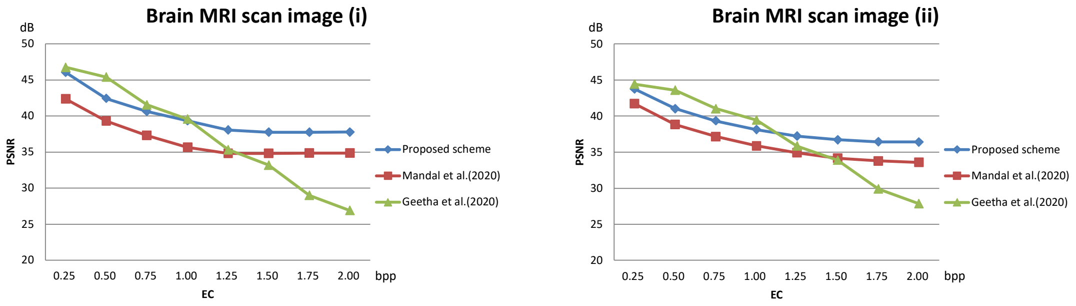


(**a**) Brain CT scan image (**i**)

(**b**) Brain CT scan image (**ii**)

(**c**) Thorax X-ray image (**i**)

(**d**) Thorax X-ray image (**ii**)



(**e**) Brain MRI scan image (**i**) (**f**) Brain MRI scan image (**ii**)

Figure 12. Comparison of PSNR on medical images; (**a**) Brain CT scan image (**i**); (**b**) Brain CT scan image (**ii**); (**c**) Thorax X-ray image (**i**); (**d**) Thorax X-ray image (**ii**); (**e**) Brain MRI scan image (**i**); (**f**) Brain MRI scan image (**ii**).

5. Conclusions

A new interpolation-based RDH scheme was proposed for hiding patient biometric information in medical images. The pixel blocks were classified into complex and smooth ones. Different methods were applied to process the two different categories. For complex blocks, the dynamic range of pixel values were preserved during data embedding. For smooth blocks, two layers of an electronic patient record can be embedded with a slight distortion in the pixel values. Compared to [20], our adaptive scheme uses the same MN interpolation and DIM embedding to ensure a high payload for the image blocks with complex textures but removes the DE embedding part of [20], which causes high distortion for the complex blocks. For the smooth image blocks, MN interpolation and DIM embedding methods cannot provide a satisfactory payload, so we adopted BM interpolation and DE embedding to ensure a good payload while preserving low distortion. From the experimental results, we can conclude that our proposed adaptive scheme can embed a satisfactory amount of biometric information with the least degree of image distortion. Our proposed adaptive scheme was also compared with state-of-the-art schemes. The performance improvement confirms the excellence of our proposed adaptive scheme.

In the application of medical image management, when the embedding capacity is large enough, the extra space can be applied to embed electronic patient records. In the future, we will focus on designing an RDH scheme that combines the hiding of patient biometric information and electronic diagnosis records.

Author Contributions: Conceptualization and methodology, H.-X.C. and J.-H.H.; software, H.-X.C.; validation, H.-X.C., J.-H.H., C.-C.C. and Y.-H.L.; writing—original draft preparation, H.-X.C. and J.-H.H.; writing—review and editing, H.-X.C., J.-H.H. and C.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chang, C.-C. Adversarial Learning for Invertible Steganography. *IEEE Access* **2020**, *8*, 198425–198435. https://doi.org/10.1109/access.2020.3034936.
2. Chi, H.; Chang, C.-C.; Liu, Y. An SMVQ compressed data hiding scheme based on multiple linear regression prediction. *Connect. Sci.* **2020**, *33*, 495–514. https://doi.org/10.1080/09540091.2020.1852179.

3. Chang, C.-C.; Li, C.-T.; Shi, Y.-Q. Privacy-Aware Reversible Watermarking in Cloud Computing Environments. *IEEE Access* **2018**, *6*, 70720–70733. https://doi.org/10.1109/access.2018.2880904.
4. Hu, Y.; Lee, H.-K.; Li, J. DE-Based Reversible Data Hiding with Improved Overflow Location Map. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *19*, 250–260. https://doi.org/10.1109/tcsvt.2008.2009252.
5. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896. https://doi.org/10.1109/tcsvt.2003.815962.
6. Coatrieux, G.; Le Guillou, C.; Cauvin, J.-M.; Roux, C. Reversible Watermarking for Knowledge Digest Embedding and Reliability Control in Medical Images. *IEEE Trans. Inf. Technol. Biomed.* **2008**, *13*, 158–165. https://doi.org/10.1109/titb.2008.2007199.
7. Liu, L.; Wang, A.; Chang, C.-C. Separable Reversible Data Hiding in Encrypted Images with High Capacity Based on Median-Edge Detector Prediction. *IEEE Access* **2020**, *8*, 29639–29647. https://doi.org/10.1109/access.2020.2972736.
8. Chi, H.-X.; Horng, J.-H.; Chang, C.-C. Reversible Data Hiding Based on Pixel-Value-Ordering and Prediction-Error Triplet Expansion. *Mathematics* **2021**, *9*, 1703. https://doi.org/10.3390/math9141703.
9. He, W.; Xiong, G.; Weng, S.; Cai, Z.; Wang, Y. Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion. *Inf. Sci.* **2018**, *467*, 784–799. https://doi.org/10.1016/j.ins.2018.04.088.
10. Li, X.; Li, B.; Yang, B.; Zeng, T. General Framework to Histogram-Shifting-Based Reversible Data Hiding. *IEEE Trans. Image Process.* **2013**, *22*, 2181–2191. https://doi.org/10.1109/tip.2013.2246179.
11. Manohar, K.; Kieu, T.D. A centroid based vector quantization reversible data hiding technique. *Multimedia Tools Appl.* **2019**, *78*, 25273–25298. https://doi.org/10.1007/s11042-019-7631-3.
12. Gao, K.; Horng, J.-H.; Chang, C.-C. A Novel (2, 3) Reversible Secret Image Sharing Based on Fractal Matrix. *IEEE Access* **2020**, *8*, 174325–174341. https://doi.org/10.1109/access.2020.3025960.
13. Bao, F.; Deng, R.; Ooi, B.; Yang, Y. Tailored Reversible Watermarking Schemes for Authentication of Electronic Clinical Atlas. *IEEE Trans. Inf. Technol. Biomed.* **2005**, *9*, 554–563. https://doi.org/10.1109/titb.2005.855556.
14. Chang, C.C.; Chou, Y.C.; Kieu, T.D. An Information Hiding Scheme Using Sudoku. In Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control, Dalian, China, 18–20 June 2008; pp. 17–22.
15. Das, S.; Muhammad, K.; Bakshi, S.; Mukherjee, I.; Sa, P.K.; Sangaiah, A.K.; Bruno, A. Lip biometric template security framework using spatial steganography. *Pattern Recognit. Lett.* **2018**, *126*, 102–110. https://doi.org/10.1016/j.patrec.2018.06.026.
16. Abadi, M.; Danyali, H.; Helfroush, M.S. Reversible Watermarking Based on Interpolation Error Histogram Shifting. In Proceedings of the 2010 5th International Symposium on Telecommunications, Tehran, Iran, 4–6 December 2010; pp. 840–845. https://doi.org/10.1109/istel.2010.5734139.
17. Luo, L.; Chen, Z.; Chen, M.; Zeng, X.; Xiong, Z. Reversible Image Watermarking Using Interpolation Technique. *IEEE Trans. Inf. Forensics Secur.* **2009**, *5*, 187–193. https://doi.org/10.1109/tifs.2009.2035975.
18. Jung, K.-H.; Yoo, K.-Y. Data hiding method using image interpolation. *Comput. Stand. Interfaces* **2009**, *31*, 465–470. https://doi.org/10.1016/j.csi.2008.06.001.
19. Lee, C.-F.; Huang, Y.-L. An efficient image interpolation increasing payload in reversible data hiding. *Expert Syst. Appl.* **2012**, *39*, 6712–6719. https://doi.org/10.1016/j.eswa.2011.12.019.
20. Mandal, P.C.; Mukherjee, I.; Chatterji, B.N. High capacity reversible and secured data hiding in images using interpolation and difference expansion technique. *Multimedia Tools Appl.* **2020**, *80*, 3623–3644. https://doi.org/10.1007/s11042-020-09341-3.
21. Geetha, R.; Geetha, S. Embedding electronic patient information in clinical images: An improved and efficient reversible data hiding technique. *Multimedia Tools Appl.* **2020**, *79*, 12869–12890. https://doi.org/10.1007/s11042-019-08484-2.
22. Lu, T.-C. Interpolation-based hiding scheme using the modulus function and re-encoding strategy. *Signal Process.* **2018**, *142*, 244–259. https://doi.org/10.1016/j.sigpro.2017.07.025.
23. Shaik, A. High-capacity reversible data hiding using 2D parabolic interpolation. *Multimedia Tools Appl.* **2018**, *78*, 9717–9735. https://doi.org/10.1007/s11042-018-6544-x.
24. Hu, J.; Li, T. Reversible steganography using extended image interpolation technique. *Comput. Electr. Eng.* **2015**, *46*, 447–455. https://doi.org/10.1016/j.compeleceng.2015.04.014.
25. Lee, C.-F.; Weng, C.-Y.; Kao, C.-Y. Reversible data hiding using Lagrange interpolation for prediction-error expansion embedding. *Soft Comput.* **2018**, *23*, 9719–9731. https://doi.org/10.1007/s00500-018-3537-7.
26. Mohammad, A.A.; Al-Haj, A.; Farfoura, M. An improved capacity data hiding technique based on image interpolation. *Multimedia Tools Appl.* **2018**, *78*, 7181–7205. https://doi.org/10.1007/s11042-018-6465-8.
27. Loan, N.A.; Parah, S.A.; Sheikh, J.A.; Akhoon, J.A.; Bhat, G.M. Hiding Electronic Patient Record (EPR) in medical images: A high capacity and computationally efficient technique for e-healthcare applications. *J. Biomed. Informatics* **2017**, *73*, 125–136. https://doi.org/10.1016/j.jbi.2017.08.002.
28. Bhardwaj, R. Hiding patient information in medical images: An encrypted dual image reversible and secure patient data hiding algorithm for E-healthcare. *Multimedia Tools Appl.* **2021**, *81*, 1125–1152. https://doi.org/10.1007/s11042-021-11445-3.
29. Jana, M.; Joardar, S.; Jana, B. A New Reversible Data Hiding Scheme by Altering Interpolated Pixels Exploiting Neighbor Mean Interpolation (NMI). In *Computational Intelligence in Pattern Recognition*; Springer: Singapore, 2022; pp. 393–402. https://doi.org/10.1007/978-981-19-3089-8_38.
30. Jana, S.; Jana, B.; Lu, T.C.; Vo, T.N. Reversible data hiding scheme exploiting center folding with fuzzy weight strategy. *J. Inf. Secur. Appl.* **2022**, *69*, 103276. https://doi.org/10.1016/j.jisa.2022.103276.