HIGH CAPACITY REVERSIBLE DATA HIDING SCHEME WITH INTERPOLATION AND THRESHOLD-BASED BIT ALLOCATION TECHNIQUE

M. Mahasree¹, N. Puviarasan², P. Aruna³

¹²Department of Computer Science and Engineering, Annamalai University, Chidambaram, Tamilnadu, India
³Department of Computer and Information Science, Annamalai University, Chidambaram, Tamilnadu, India
¹mahasree05@gmail.com, ²npuvi2410@yahoo.in, ³arunapuvi@yahoo.co.in

Corresponding Author: M. Mahasree

https://doi.org/10.26782/jmcms.2020.07.00062

Abstract

Secret communication through lossless data hiding techniques is an active research field where payload management is a challenging task. The tradeoff between stego quality and payload capacity generally exists in such fields. To achieve higher payload, interpolation based data hiding techniques (IRDH) are opted in several areas including e-governance, military imagery, medical imaging systems etc. The purpose of interpolation in hiding systems is to provide better hiding capacity without altering the original pixels. Conventional interpolation-based hiding techniques lack in providing high embedding capacity due to some restrictions in embedding rules. Thus, this paper encompasses an effective embedding procedure for interpolation based reversible data hiding schemes to fulfill the capacity requirement. The objective of our proposed scheme is to increase the payload capacity by making use of all interpolated pixels in the cover image with good visual quality. Particularly, the proposed Threshold-based Bit Allocation (TBA) technique efficiently assigns the number of bits that can be embedded in an interpolated pixel. Experimental results show that the proposed interpolation based reversible data hiding technique performs better than many state-of-the-art methods in terms of hiding capacity as well as visual quality.

Keywords: Lossless Data Hiding, Interpolation based Reversible Data Hiding (IRDH), payload capacity, stego quality, Threshold-based Bit Allocation (TBA)

I. Introduction

Data Hiding has been used to securely share sensitive and private information over internet [XI]. Data Hiding is categorized into two types. One is Reversible data hiding (RDH) and the other is Irreversible data hiding (IDH) [IV]. Recently, Interpolation-based Reversible Data Hiding (IRDH) Schemes are being developed for...
high embedding capacity. Interpolation is one of the image processing techniques, used to enlarge an image by creating new pixels [VIII]. It exploits the dependency of neighborhood pixels to create a new pixel. The most common interpolation techniques used in imaging are Nearest Neighbor Interpolation (NNI) and Bilinear Interpolation (BI). But the scaled up images obtained from these traditional methods does not satisfy the image quality needed for data hiding schemes [XIV]. So, various interpolation schemes like Neighbor Mean Interpolation (NMI), Interpolation by Neighboring Pixels (INP), Enhanced Neighbor Mean Interpolation (ENMI) etc., have been developed for reversible data hiding [XIV], [IX]. However, these methods suffer from capacity limitations. The interpolated pixels are not effectively utilized for data embedding which leads to low embedding capacity. Therefore, there is possibility for increasing the payload.

In this paper, a high capacity interpolation based reversible data hiding is proposed. Neighbor Mean Interpolation is used to interpolate the given image. This up-scaled image is the cover image. Based on a threshold value, the number of bits to be embedded in each interpolated pixel is decided by the proposed Threshold-based Bit Allocation (TBA) technique.

The remainder of this paper is organized as follows. Section 2 discusses the related works in IRDH. Section 3 presents the proposed system in detail. Section 4 shows the experimental results. Conclusions are provided in Section 5.

II Related Work

Jung and Yoo first introduced the use of interpolation in reversible data hiding. They proposed an interpolation method namely Neighbor Mean Interpolation (NMI) to embed secret data into those interpolated pixels [VI]. Lee and Huang presented a better interpolation method to NMI which is named as Interpolation by Neighboring Pixels (INP). This method provides better visual quality then Jung and Yoo’s method [XII]. Tang et al. proposed a better RDH using interpolation to increase the hiding capacity [X]. Hu and Li extended the INP method using Interpolation by Maximizing the difference values between Neighboring Pixels (IMNP). This technique improved the embedding capacity of Lee and Huang’s method [III]. Lu proposed an interpolation based lossless hiding using message recoding step. Here, INP interpolation is used. The re-encoding scheme assigns smaller codes to the secret message based on its frequency of occurrence [VIII].

Mohammad et al. presented a new and computationally simple reversible scheme based on interpolation. In this work, the number of embeddable bits and stego quality is maintained by threshold value [XV]. Luo et al. proposed a reversible watermarking scheme based on interpolation-error expansion. The additive interpolation is followed by difference expansion of the interpolated pixels [II]. Wang et al. exploited the interpolation technique and direction order mechanism. Interpolation errors are calculated for one type of pixels called wall pixels whereas difference values are calculated for other type of pixels called non-wall pixels. Embedding is carried out based on direction order and histogram shifting [I]. Biswapati et al. proposed a reversible data hiding scheme based on a weighted matrix. Modulus function and
element-wise multiplication steps are employed to hide secret data [X]. Govind and Wilsry presented a new RDH with directional interpolation and difference expansion using a threshold value [XIII].

III Proposed Work

In this section we present our novel reversible data hiding using interpolation and Threshold-based Bit Allocation (TBA) methods. The general structure of interpolation based reversible data hiding scheme is shown in Fig. 1. The proposed RDH proceeds by the application of NMI method. Then, the interpolated pixels are processed in 3 x 3 blocks. Based on the difference value between interpolated pixels and neighbor pixels, payload capacity of each interpolated pixel is calculated using TBA strategy. Then the bits are embedded into the interpolated pixels without degrading the overall visual quality of the stego image. Next, the extraction algorithm is explained where both cover and secret data are separated from the stego image reversibly.

![Fig. 1 Framework for Interpolation-based Reversible Data Hiding Techniques](image)

**Neighbor Mean Interpolation (NMI)**

Image interpolation is a technique by which low resolution image can be scaled up to high resolution image without degrading the visual quality. There are many interpolation techniques used in image hiding. The first proposed interpolation technique for RDH is Neighbor Mean Interpolation (NMI). This method is used in this paper because of the following reasons.

- NMI is computationally simple
- NMI provides better visual quality than other interpolation techniques

Let \( O(i,j) \) be the original image of size 256 x 256 and \( C(i,j) \) be the cover image of size 512 x 512 obtained from original image using NMI interpolation. The scaling procedure is described as follows. As pointed out in Fig. 2, for each 2 x 2 block of
original image, five new pixels are inserted to form a 3 x 3 block. The corner pixels of the cover image \( C(0,0), C(0,2), C(2,0) \) and \( C(2,2) \) are the unaltered original image pixels \( O(0,0), O(0,1), O(1,0) \) and \( O(1,1) \) respectively as shown in Fig. 2. The middle pixels are calculated as follows:

\[
\begin{align*}
C(0,1) &= \lfloor (C(0,0) + C(0,2))/2 \rfloor \\
C(1,0) &= \lfloor (C(0,0) + C(2,0))/2 \rfloor \\
C(1,1) &= \lfloor (C(0,0) + C(0,1) + C(1,0))/3 \rfloor \\
C(1,2) &= \lfloor (C(0,2) + C(2,2))/2 \rfloor \\
C(2,1) &= \lfloor (C(2,0) + C(2,2))/2 \rfloor \\
\end{align*}
\] (1)

Fig. 2 Generation of cover image from original image using NMI

**Scope of Improvement**

In the embedding scheme of other existing methods, the number of secret bits to be embedded into the interpolated pixels \( C(i, j+1), C(i+1, j) \) and \( C(i+1, j+1) \) are respectively calculated using Eq. 2 [VI], [XII], [V], [III], [XV].

\[
n_k = \lfloor \log_2 |d_k| \rfloor, \quad k = 1,2,3
\] (2)

where \( d_k \) depicts the relationship between the interpolated pixels and the unchanged pixel. This technique allocates number of secret bits equal to the \( n_k \) value and achieves good visual quality. However, this technique does not utilize all the interpolated pixels of the cover image as explained below.

\[
n_k = \begin{cases} 
\infty, & \text{if } d_k = 0 \\
0, & \text{if } d_k = 1 
\end{cases} \quad k = 1,2,3
\] (3)

For these values, no secret bit is embedded into the respective pixel leading to low embedding capacity. Table 1 shows how many interpolated pixels are left without embedding in existing data embedding technique [VI]. The cover image is composed of 196608 interpolated pixels. Of which, 93828 pixels are left unused.
To solve this issue, new improvements are proposed as follows.

- The interpolated pixels with \( n_k = \infty \) and \( n_k = 0 \) are also used for embedding.
- When the value of \( n_k \) is higher, more bits are embedded into a single pixel which may affect visual quality. Hence, the number of bits is restricted to a threshold value such that bits are optimally distributed.

**Embedding using Threshold-based Bit Allocation (TBA)**

After interpolation, the next step is embedding. The cover image is processed in 2 x 2 non-overlapping blocks. At each block, the first pixel \( C(i,j) \) is kept unchanged for distortion-less recovery. The remaining pixels \( C(i,j+1), C(i+1,j) \) and \( C(i+1,j+1) \) are used for embedding. The number of bits to be embedded is calculated using two values \( d_k \) and \( l_k \). The difference between each interpolated pixel and the mean of 2 x 2 block is denoted as \( d_k \).

\[
M = \lfloor (C(i,j) + C(i,j+1) + C(i+1,j) + C(i+1,j+1))/4 \rfloor \tag{4}
\]

\[
d_1 = |C(i,j+1) - M| \\
\]

\[
d_2 = |C(i+1,j) - M| \\
\]

\[
d_3 = |C(i+1,j+1) - M| \tag{5}
\]

The log base 2 values of \( d_k \) are denoted as \( l_k \). The equations are as follows.

\[
l_1 = \lfloor \log_2(d_1) \rfloor \\
l_2 = \lfloor \log_2(d_2) \rfloor \\
l_3 = \lfloor \log_2(d_3) \rfloor \tag{6}
\]
The proposed threshold-based Bit Allocation technique equally distributes the secret bits to be embedded into interpolated pixels. The number of bits that can be embedded into pixel is given by the equations as follows.

\[
 n_k = \begin{cases} 
    t_h, & \text{if } l_k \in \{\infty, 0,1\} \\
    t_h + 1, & \text{if } l_k \in \{2,3,4\} \\
    t_h + 2, & \text{if } l_k \in \{5,6,7\}
\end{cases}
\]  

(7)

where \(k=1,2,3\) denotes the interpolated pixels \(C(i,j+1), C(i+1,j)\) and \(C(i+1,j+1)\) respectively. Thus, the value of \(n_k\) denotes the length of secret bits. Usually this value is converted into decimal number and added with the cover pixels to form the stego pixels. But this may lead to overflow condition where the stego pixel value exceeds the maximum intensity of 255. So, to overcome this situation, limitation variable \(R\) is introduced.

\[
 R = 255 - (2^{(t_h+2)} - 1)
\]

(8)

Let \(s_k\) be the bit string of length \(n_k\) with \(k=1, 2, 3\). The embedding equations are modified as follows.

\[
 S(i,j) = C(i,j)
\]

\[
 S(i,j + 1) = \begin{cases} 
    C(i,j + 1) + s_1, & \text{if } C(i,j + 1) \leq R \\
    C(i,j + 1) - s_1, & \text{otherwise}
\end{cases}
\]

\[
 S(i+1,j) = \begin{cases} 
    C(i+1,j) + s_2, & \text{if } C(i+1,j) \leq R \\
    C(i+1,j) - s_2, & \text{otherwise}
\end{cases}
\]

\[
 S(i+1,j+1) = \begin{cases} 
    C(i+1,j+1) + s_3, & \text{if } C(i+1,j+1) \leq R \\
    C(i+1,j+1) - s_3, & \text{otherwise}
\end{cases}
\]

(9)

**Algorithm 1. Embedding**

**Input:** An Original Image \(O\) sized \(r \times c\) and secret bit stream \(s\)

**Output:** A Stego Image \(S\) sized \(2r \times 2c\)

**Step 1:** Resize the original image \(O\) using NMI Interpolation and obtain cover image \(C\) of size \(2r \times 2c\)

**Step 2:** Divide Cover into 2 x 2 non-overlapping blocks

**Step 3:** Find the mean of the block denoted as \(M\) using Eq. 4

**Step 4:** Calculate the absolute difference between the interpolated pixels and \(M\) as given in Eq. 5

**Step 5:** Proceed with binary logarithm for \(d_k\) which is denoted as \(l_k\)

**Step 6:** The length of secret bits is denoted as \(n_k\) and calculated using Eq. 7. The threshold value \(th\) is the user defined value to choose between stego quality and capacity.
Step 7: Check the overflow condition using Eq. 8. Convert the binary secret bits into decimal.

Step 8: According to Eq. 9 embed the secret bit strings into cover image to form a stego image.

**Extraction**

At the extraction side, the original image is separated from the cover image by collecting the unchanged pixel $S(i, j)$ in every 2 x 2 block. Now, same NMI Interpolation is performed on recovered original image. By finding the difference matrix between the interpolated image and the stego image, secret bit stream is recovered losslessly.

**Algorithm 2. Extraction**

**Input:** Stego Image $S$ sized $2r \times 2c$

**Output:** Original Image $O$ sized $r \times c$ and secret bit stream $s$

**Step 1:** Divide Stego into 2 x 2 non-overlapping blocks. Collect all the first pixels $S(i, j)$ from each block to get the Original Image $O$.

**Step 2:** Use the NMI interpolation to get the interpolated image from the recovered Original image (Eq.1).

**Step 3:** Find the absolute difference between the interpolated image and the Stego image.

**Step 4:** The difference values is the value of integer hidden in stego image.

**Step 5:** Follow the steps of embedding (Eq. 4, 5, 6) to find the $n_k$ value for each block. Convert the decimal into binary bits of length $n_k$ using Eq. 7.

**Step 6:** Extract all the secret bits to regain the secret data losslessly.

**Auxiliary information**

The auxiliary information needed for extraction are the length of total secret bits embedded and the threshold value. To represent the secret data length, we need a maximum of 20 bits. The maximum value of possible threshold value is 5 and hence it requires 3 bits in binary form. So, the additional data of $L_{TH} + L_S$ is embedded into the first two blocks of cover image using LSB substitution as shown in Fig. 3. Then, the proposed embedding starts with the third block and proceeds till all required bits are hidden successfully.
IV. Experimental Results

This section presents the experimental results and comparison of the proposed method with other state-of-art methods. Experiments were done on 8-bit grayscale images of size 512 x 512. These images serve as ground truth. The interpolated cover image is compared with ground truth (input image) image to show the performance of the interpolation scheme. So, the original image (carrier of secret data) is formed by reducing the input image to ¼ of its size. It is to be noted that for all the existing methods taken for comparison and the proposed method we used Bicubic Interpolation for this reduction step.

The image quality of the embedded image can be evaluated using PSNR as follows:

\[
PSNR = 10 \times \log \left( \frac{255^2}{MSE} \right)
\]

\[
MSE = \frac{1}{MN} \sum_{i=0}^{M} \sum_{j=0}^{N} (S(i, j) - I(i, j))^2
\]

Fig. 3 Embedding of Auxiliary information using LSB substitution in first two 2 x 2 blocks

where PSNR is Peak Signal-to-Noise Ratio and MSE is Mean Square Error between Stego image S(i,j) and Input image I(i,j) with dimension M x N. Structural Similarity Index Measure is given by,

\[
SSIM(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{\mu_x^2 + \mu_y^2 + c_1(\sigma_x^2 + \sigma_y^2 + c_2)}
\]

\[
c_1 = (k_1L)^2, \quad c_2 = (k_2L)^2
\]

where \(\mu_x\) and \(\mu_y\) are average values; \(\sigma_x^2\) and \(\sigma_y^2\) denotes variance, \(\sigma_{xy}\) is the covariance of \(x\) and \(y\); \(c_1\) and \(c_2\) are variables for stabilization, \(L=255\) denotes pixel range, \(k_1=0.01, k_2=0.03\) are default constants and \(x\), \(y\) are stegeo and input images respectively. Embedding capacity denotes the total number of bits that can be embedded into the cover image. It is named as BPP (bits per pixel). The secret data are the randomly generated binary bits.
Table 2 shows how the proposed system is efficient when compared to existing method [VI]. It is evident that higher capacity is due to the increase in number of pixels with $n_k$ values 0 and infinity. Minimum bits are selected for those pixels with higher occurrences of $n_k$ to maintain the stego image visibility.

**Table 2:** Comparison of Proposed method with existing IRDH [VI] for Lena Image

| Range of $n_k$ | Jung and Yoo [VI] | Proposed TBA |
|---------------|-------------------|--------------|
|               | Number of occurrences of $n_k$ | Number of bits embedded for given $n_k$ | Total Capacity (in bits) | Number of occurrences of $n_k$ | Number of bits embedded for given $n_k$ | Total Capacity (in bits) |
| $\infty$     | 36835             | 0             | 0               | 96286             | 288858               | 507429               |
| 0             | 56993             | 0             | 0               | 42646             | 127938               |
| 1             | 50085             | 50085         | 16037           | 30211             | 90633                |
| 2             | 28502             | 57004         | 190540          | 16037             | 64148                |
| 3             | 15309             | 45927         | 8175            | 32700             |
| 4             | 7009              | 28036         | 190540          | 11436             |
| 5             | 1762              | 8810          | 394             | 1970              |
| 6             | 113               | 678           | 0               | 0                 | 1970                 |
| 7             | 0                 | 0             | 0               | 0                 |

Table 3 shows the performance of different threshold values for the proposed TBA method. The value of $t_h$ is restricted between 1 and 3 to balance the tradeoff between capacity and PSNR. Thus the user can define the $t_h$ value according to their need.

Table 4 shows the PSNR and capacity comparison of the proposed scheme with other existing techniques. The capacity of the proposed technique is higher than all other methods. Despite this increment, our PSNR values are much closer to other existing techniques with lower capacity. This proves the novelty of the proposed method.
Table 3: Performance of the proposed method with different thresholds

| Cover Images | Proposed \( th=1 \) | Proposed \( th=2 \) | Proposed \( th=3 \) |
|--------------|-------------------|-------------------|-------------------|
|              | PSN R  | BP P  | SSIM  | PSN R  | BP P  | SSIM  | PSN R  | BP P  | SSIM  |
| Barbara      | 31.85  | 0.94  | 0.798 | 5      | 31.74  | 1.69  | 0.792 | 7      | 31.22  | 2.44  | 0.766 | 7      |
| Goldhill      | 32.76  | 0.91  | 0.846 | 1      | 32.58  | 1.66  | 0.839 | 5      | 31.86  | 2.41  | 0.810 | 4      |
| Jetplane      | 34.90  | 0.86  | 0.923 | 8      | 34.61  | 1.61  | 0.915 | 0      | 33.40  | 2.36  | 0.879 | 3      |
| Lake          | 32.50  | 0.91  | 0.855 | 5      | 32.35  | 1.66  | 0.849 | 0      | 31.69  | 2.41  | 0.822 | 4      |
| Baboon        | 29.90  | 1.03  | 0.721 | 8      | 29.86  | 1.78  | 0.717 | 7      | 29.66  | 2.53  | 0.702 | 7      |
| Lena          | 34.57  | 0.85  | 0.899 | 5      | 34.28  | 1.60  | 0.891 | 7      | 33.14  | 2.35  | 0.859 | 1      |
| Peppers       | 33.67  | 0.84  | 0.850 | 3      | 33.44  | 1.59  | 0.843 | 1      | 32.55  | 2.34  | 0.812 | 8      |
| Boat          | 33.26  | 0.89  | 0.879 | 0      | 33.08  | 1.64  | 0.871 | 1      | 32.33  | 2.39  | 0.838 | 9      |

Table 4: Comparison of the proposed method with existing techniques

| Cover Images | Jung and Yoo [VI] | Lee and Huang [XII] | Tang et al. [V] | Mohammad et al. [XV] | Proposed |
|--------------|-------------------|---------------------|-----------------|-----------------------|----------|
|              | PSN R  | BP P  | PSN R  | BP P  | PSN R  | BP P  | PSN R  | BP P  | PSN R  | BP P  |
| Barbara      | 31.84  | 1.15  | 31.69  | 1.84  | 31.44  | 2.27  | 31.33  | 1.45  | 31.22  | 2.44  |
| Goldhill     | 32.75  | 0.92  | 32.46  | 1.63  | 32.12  | 2.09  | 32.09  | 1.26  | 31.86  | 2.41  |

Copyright reserved © J. Mech. Cont.& Math. Sci.  
M. Mahasree et al
Another set of experiments were conducted with fingerprint images as secret data. The dimensions of fingerprint images are taken as 256 x 256. This is shown in Fig. 4. The original image is 256 x 256 in size. Thus, an equal sized secret image can be reversibly hidden within another image of same size using interpolation. The fingerprint images used in experiment are shown in Fig 5. Table 5 shows the performance of the proposed technique with fingerprint images as secret data. Here the threshold value is set to 3 to attain the needed BPP value 2. Since the secret image has dimension 256 x 256, the bits per pixel will be its total count, 256 * 256 * 8 = 524288, which is divided by the size of cover image, 524288 / ( 512 * 512 ) = 2 BPP.

![Block diagram of the proposed IRDH for embedding fingerprint image into cover image](image)

**Fig. 4** Block diagram of the proposed IRDH for embedding fingerprint image into cover image

---

**Copyright reserved © J. Mech. Cont.& Math. Sci. M. Mahasree et al**
Fig. 5 Fingerprint Images as secret data

Table 5: Performance of the proposed method with fingerprint image as secret data

| Cover Images | Image 1 | Image 2 |
|--------------|---------|---------|
|              | PSNR    | SSIM    | PSNR    | SSIM    |
| Barbara      | 31.23   | 0.7688  | 31.14   | 0.7662  |
| Goldhill     | 31.84   | 0.8121  | 31.73   | 0.8091  |
| Jetplane     | 33.28   | 0.8809  | 33.09   | 0.8777  |
| Lake         | 31.68   | 0.8241  | 31.57   | 0.8214  |
| Baboon       | 29.67   | 0.7050  | 29.64   | 0.7031  |
| Lena         | 33.03   | 0.8594  | 32.85   | 0.8564  |
| Peppers      | 32.46   | 0.8136  | 32.32   | 0.8109  |
| Boat         | 32.25   | 0.8391  | 32.12   | 0.8361  |

V Conclusion

In this paper, a novel Threshold-based Bit Allocation is proved to be efficient for interpolation based information hiding. Neighbor Mean Interpolation is preferred for up-scaling the cover image. Then, the proposed embedding schemes allowed high payload to be embedded inside the cover image without disturbing the original pixels. Proposed method has an advantage of not using any location map or secret pin to be shared prior to communication. Also, the extraction of cover image is blind technique i.e. the cover image is extracted without any procedures or additional information. Experimental results showed the efficiency of the proposed technique suitable in medical, fingerprint, military and other intelligent security services where secret object as well as cover object is equally important. PSNR, SSIM and BPP values also demonstrated the efficiency of the proposed TBA based reversible information hiding techniques.
References

I. A. A. Mohammad, A. Al-Haj, M. Farfoura, "An improved capacity data hiding technique based on image interpolation", Multimedia Tools and Applications, Springer, Vol.: 78, Issue: 6, pp. 7181-7205, 2019

II. A. Malik, G. Sikka, H. K. Verma, "Image interpolation based high capacity reversible data hiding scheme", Multimedia Tools and Applications, Springer, Vol.: 76, Issue: 22, pp. 24107-24123, 2017

III. C. Lee, Y. Huang, "An efficient image interpolation increasing payload in reversible data hiding", Expert Systems with Applications, Elsevier, Vol.: 39, Issue: 8, pp. 6712-6719, 2012

IV. J. Biswapati, G. Debasis, M. S. Kumar, "Weighted Matrix Based Reversible Data Hiding Scheme Using Image Interpolation", International Conference on Computational Intelligence in Data Mining (CIDM), Vol.: 2, pp. 239-248, 2015

V. J. Hu, T. Li, "Reversible steganography using extended image interpolation technique", Computers & Electrical Engineering, Elsevier, Vol.: 46, pp. 447-455, 2015

VI. K. Jung, K. Yoo, "Data hiding method using image interpolation", Computer Standards & Interfaces, Elsevier, Vol.: 31, Issue: 2, pp. 465-470, 2009

VII. L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, "Reversible Image Watermarking Using Interpolation Technique", IEEE Transactions on Information Forensics and Security, Vol.: 5, Issue: 1, pp. 187-193, 2010

VIII. M. A. Wahed, H. Nyeem, "Reversible data hiding with interpolation and adaptive embedding", Multimedia Tools and Applications, Springer, Vol.: 78, Issue: 8, pp. 10795-10819, 2019

IX. M. Mahasree, N. Puviarasan, P Aruna, "Pixel Value Ordering Based Reversible Data Hiding with Novel MPBS Strategy", International Journal of Engineering and Advanced Technology (IJET), Vol.: 9, Issue: 3, pp. 1518-1524, 2020

X. M. Tang, J. Hu, W. Song, "A high capacity image steganography using multi-layer embedding", Optik, Elsevier, Vol.: 125, Issue: 15, pp. 3972-3976, 2014

XI. P. V. S. Govind, M. Wilsey, "A New Reversible Data Hiding Scheme with Improved Capacity Based on Directional Interpolation and Difference Expansion", International Conference on Information and Communication Technologies (ICICT 2014), pp. 491-498, 2015

XII. S. Meikap, B. Jana, "Directional PVO for reversible data hiding scheme with image interpolation", Multimedia Tools and Applications, Springer, Vol.: 77, Issue: 23, pp. 31281-31311, 2018
XIII. T. Lu, "Adaptive (k, F1) interpolation-based hiding scheme", Multimedia Tools and Applications, Springer, Vol.: 76, Issue: 2, pp. 1827-1855, 2017

XIV. T. Lu, "An interpolation-based lossless hiding scheme based on message recoding mechanism", Optik, Elsevier, Vol.: 130, pp. 1377-1396, 2017

XV. X. Wang, C. Chang, T. Nguyen, M. Li, "Reversible data hiding for high quality images exploiting interpolation and direction order mechanism", Digital Signal Processing, Elsevier, Vol.: 23, pp. 569-577, 2013