Directional pixel value ordering based secret sharing using sub-sampled image exploiting Lagrange polynomial

Sudipta Meikap¹ · Biswapati Jana²

© Springer Nature Switzerland AG 2019

Abstract

In this paper, we proposed secret sharing within the sub-sampled image using Directional Pixel Value Ordering (DPVO) for Reversible Data Hiding scheme. The original cover is partitioned into sub-sampled images which extend its size through image interpolation technique. The secret information has been converted through Lagrange interpolation polynomial techniques. These new secret data is embedded within pixels of each interpolated sub-sampled images using DPVO. In the decoder side, the secret data is taken out from the pixels of every stego image using reverse DPVO. After that, Lagrange's interpolation is applied to generate the original secret message. The proposed scheme enhanced the security for the sharable characteristics of secrets amongst several images. It raises the data embedding capacity and improves the visual quality that is determined by peak signal to noise ratio (PSNR). The average PSNR value is over 60 dB. It shows the superiority of the new algorithm over the other existing data hiding methods regarding payload of data, quality, and security of the image. Also, the stego images have evaluated through standard deviation, regular singular analysis, correlation coefficient, normalized cross-correlation and structural similarity index in between original cover and marked (stego) image to display the robustness of our result among the several steganographic strikes.

Keywords Directional pixel-value-ordering · Image interpolation · Reversible data hiding · Sub-sampled image · Embedding capacity · RS analysis · Structural similarity index · Normalized cross-correlation · Steganography

1 Introduction

In the domain of information security, the steganography technique has the prime role to communicate innocent secret data in between sender and receiver for both academic and industrial researchers. This technique is classified in two ways: irreversible data hiding and reversible data hiding. In irreversible data hiding, the original image can only hide the secret information. It can not recover any secret data. Whereas, both the secret data can be hidden into the image and get back from the marked image in Reversible Data Hiding (RDH). So, RDH is generally used

in the areas of medical and military images, remote sensing and copyright protection etc. In the RDH schemes two important parameters are evaluated: embedding distortion and embedding capacity (EC). In general, the prime challenges of effective RDH scheme is to enhance the quality of an image as well as data hiding capacity.

Over the past years, different RDH schemes have been proposed by the researchers. Applying method of difference expansion (DE) Tian [27] proposed a technique to conceal secret data into the pixel pair. Later, Alattar [1] applied four pixels differentiation by revising the Tian's method. Ni et al. [20] presented the RDH technique which

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s42452-019-0659-1>) contains supplementary material, which is available to authorized users.

✉ Biswapati Jana, biswapatijana@gmail.com; Sudipta Meikap, sudiptameikap@gmail.com | ¹Department of Computer Science, Hijli College, Paschim Medinipur, West Bengal 721306, India. ²Department of Computer Science, Vidyasagar University, Midnapore 721102, India.

SN Applied Sciences (2019) 1:645 | <https://doi.org/10.1007/s42452-019-0659-1>

Received: 19 February 2019 / Accepted: 21 May 2019 / Published online: 27 May 2019

uses the minimum histogram. After that, Lin et al. [14] and Tsai et al. [28] presented multilevel RDH scheme using modification of histogram. Kim et al. [8] described a technique through correlation among the sub-sampled images in 2009. For example, in Lena image, the embedding capacity and image quality in PSNR of Kim et al.'s. [8] were 20,121 data bits and 48.9 dB respectively. After that, Luo et al. [17] presented a scheme by selection of the median pixel used as sub-image reference in each block which is partitioned in four section to keep the block median value. The PSNR value was 48.9 dB and the payload in bits was 0.11 bpp.

Li et al. [13] presented a technique using pixel value ordering(PVO) where only the maximum and minimum values are changing due to the data embedding in 2013. A technique where all pixels within a block are sorted in rising up fashion and modified the value of the maximal or minimal pixel for data embedding is called Pixel Value Ordering (PVO). The payload and quality of image were 32000 bits and 59.8 PSNR dB (for EC of 10,000 bits) respectively for Lena image. Lee et al. [11] presented two staged multilevel RDH scheme using Lagrange interpolation. By using Lagrange interpolation they generate predicted image. Then image difference is computed and hide the secret information by using histogram shifting algorithm. In this scheme, they construct one marked image with various embedding capacity. For example, in Baboon image, the embedded data bits was 0.88 bpp and the quality of image in PSNR was 48.32 dB. Peng et al. [24] improved both visual quality as well as embedding capacity where more smooth blocks were used. The Embedding capacity and image quality were 38,000 bits and PSNR 60.4 dB (for payload of 10,000 bits) respectively for Lena image. In 2015, Qu and Kim [25] presented a RDH technique through pixel-based PVO where the quality of image in PSNR and payload were 60.3 dB (for payload 10,000 bits) and 46,000 bits respectively for Lena image. In 2016, Ou et al. [21] presented a technique where improved PVO was used through modification of several histograms. The reversible data hiding scheme based on dual image has been introduced by some researchers [5, 7]. Secret message hiding technique using sub-sampled images has been developed by Jana [6] in 2017. According to him, this hiding process is done through Lagrange's interpolation polynomial on sub-sampled images. In this techniques, these images have been made from the cover image. Then, these are interpolated by interpolation techniques. Take any secret message and apply it with Lagrange interpolation function $f(x)$. This $f(x)$ value is converted to the binary value of 12 bits. This 12 bit binary data is partitioned into four fragment. Embed these four fragment by adding its corresponding value at position (1, 2), (2, 1), (2, 3) and (3, 2) of size (3×3) interpolated image block. The value of position

(2, 2) in interpolated image block is added with the value of x in $f(x)$.

In data extraction, the difference of pixel value is calculated from marked image independently. Then collect the difference pixel value and value of x in $f(x)$. These values are combined to create the $(x, f(x))$ value. Then applying Lagrange recover function to extract the secret data. For Lena image, the payload and quality through Jana [6] scheme were 130,000 bits and 50.60 PSNR dB respectively. Some researcher has been focused on interpolation based data hiding techniques [2, 10, 12, 15, 16, 18, 29, 32–37]. [23] and [3] are focused on watermarking and image authentication based data hiding techniques respectively.

Performance of security enhancement with RDH while keeping better quality of image using several images through Directional Pixel Value Ordering (DPVO) is still an important research matter. When hidden data is embedded using PVO technique through different direction [19] (i. e. horizontal, vertical and diagonal) one after other is called Directional PVO. We propose a safe RDH technique using Lagrange's interpolation polynomial for sub-sampled image using DPVO. Here, any user can conceal hidden data bits in between sub-sampled images. Latter, the user can get back hidden data bits from marked images.

It is difficult task to raise payload with unalter image quality through numerous process of data embedding in the different direction on image block. So, we consider data hiding process is applied through firstly horizontal, secondly vertical and lastly diagonal directions. The hidden message is embedded into the first, second, third and so on minimum and maximum pixels in every direction. So, payload has been increased five times than Jana's [6] scheme and the visual quality is raised than Jana's [6] scheme. For solving lower order than higher order polynomials, it is useful to apply Lagrange interpolation. Here, sub-sampled image is four ($h = 4$) and order is $(h - 1)$ of function $f(x)$. For some areas, where enhanced payload and better quality of the image are necessary, our proposed scheme is useful in that area.

Secret sharing is a method where a secret message is distributed between a group of members and each member carries a share of a secret message. The secret information can be rebuilt only when a sufficient number(h) of shares are combined together and no $(h - 1)$ can do so. Secret data communication using DPVO on several sub-sampled images through Lagrange interpolation polynomial raises the level of security. First, the hidden message is converted to its ASCII value and then compute the value of Lagrange's interpolation function with parameters: threshold, coefficient, number of trusted parties and size of l prime. The function value produces another level of security on messages. Then, the function values are converted to its corresponding binary bit stream. These

values are distributed between sub-sampled images with block size. Only the number(threshold value) of images are needed to retrieve the information. If the hackers want to hack the confidential information, they must need the said parameters which are hard to guess simultaneously. It is also unable to get back the hidden message from the marked images less than the threshold value.

In Sect. 2, we describe the proposed scheme along with an example. Experimental results, comparisons and some steganographic attacks are given in Sect. 3. Finally, Sect. 4 describes the conclusions of this paper.

1.1 Motivation:

Our prime motivations are described as follows:

- (i) Secret data concealing through Lagrange interpolation polynomial was irreversible. Using Directional PVO (DPVO) method within interpolated sub-sampled images, it is possible to insert and take out secret message to cover image and from marked (stego) image respectively.
- (ii) So far, there are several data hiding methods introduced by use of one or two images. It is complicated task to embed and extract hidden message with security among multiple images. So, we have introduced data hiding scheme within interpolated sub-sampled images which are served as numerous images for improve visual quality, data security. This scheme also accomplish reversibleness.
- (iii) Use of the Lagrange interpolation function, it produces new secret data bit from the original secret message which is distributed among the interpolated sub-sampled images. This lead to enhance the security. Using both DPVO and Lagrange function several unknown parameters (i. e. image block size, constant coefficient value and order of function) which are impossible to guess for the adversary that can improve security.
- (iv) So far, the stego image quality is calculated in PSNR dB through RDH schemes within sub-sampled images using Lagrange interpolation polynomial was limited because of modifying the pixel by large value. We solve it by using DPVO. So that, our proposed scheme modifies a pixel by 0 or 1 value which leads to improve visual quality of an image.
- (v) If any sub-sampled stego image destroys or losses through transmission, then it is possible to retrieve the secret message successfully from rest of other stego images. It is mentioned that, there are predefined stego images required to recover the hidden data which is dependent on lagrange function.

- (vi) So far, data embedding capacity (EC) within sub-sampled images using Lagrange interpolation polynomial was limited. Our proposed scheme is able to be done to improve payload of secret data bit and also visual quality which are essential in many application like medical and military image.

2 Proposed method

In this section, the process of data bit embedding and extraction through secret shares within sub-sampled images using Directional Pixel Value Ordering is discussed. The proposed method includes three stages: Initialization stage, Embedding stage and Extraction stage. This procedure is described below:

2.1 Initialization stage

At first, we take a cover image. Assume that $(P \times Q)$ is an image size indicated by $CI(a, b)$, where the range of $a = 0, \dots, (P - 1)$ and the range of $b = 0, \dots, (Q - 1)$. We also assume that, Δu in a row direction and Δv in a column direction are two components of sampling denoted as the expected intervals of sub-sampling. The sub-sampled of an image(2-D) is at uniform intervals shown in Fig. 1. It is known as the sub-sampling process.

In this paper, we use only four sub-sample images. From original cover image, we create four sub-sampled images (Sl_1, Sl_2, Sl_3, Sl_4) of size $(a/2 \times b/2)$ by using following equation 1.

$$\begin{aligned}
 Sl_1(rows, cols) &= CI(2 * rows, 2 * cols) \\
 Sl_2(rows, cols) &= CI(2 * rows, 2 * cols + 1) \\
 Sl_3(rows, cols) &= CI(2 * rows + 1, 2 * cols) \\
 Sl_4(rows, cols) &= CI(2 * rows + 1, 2 * cols + 1)
 \end{aligned} \tag{1}$$

where $rows = 0, 1, \dots, (a/2) - 1$ and $cols = 0, 1, \dots, (b/2) - 1$. After creation of four sub-sampled images, we enlarge each of the sub-sampled image using interpolation technique by computing the average of neighbor pixels shown in the following Fig. 2. The interpolate row and column is dependent on original image block size. For

Fig. 1 Block diagram of sub-sampled image block

Fig. 2 Block diagram of interpolated sub-sampled image block. Green and pink color represents interpolate row and column respectively

example, if the original block size is $(w \times w)$ then $(w - 1)$ will be the interpolate row and column. The interpolated sub-sampled image will be $((2w - 1) \times (2w - 1))$. To make it $(2w \times 2w)$, added one column and one row at the image end through copied the same pixel values of $(2w - 1)^{th}$ column and row.

The secret message can be lost if there is any displacement of the sampled image during transmission from sender to receiver or vice versa. To get back the message fixed the (n) number sub-sampled marked image which carries the secret message and only h of them to reconstruct the message, but no $(h - 1)$ of them can do so, where h is the threshold value. Here, we select three amongst four which cases to retrieve the hidden data required any three sub-sampled marked images amongst four marked images. Now, we consider that there is eight bits secret message (SM). After that apply Lagrange interpolation function $f(x)$ with threshold (h) is described below:

$$f(x) = a_0 + a_1x + \dots + a_{(h-2)}x^{(h-2)} + a_{(h-1)}x^{(h-1)} \quad (2)$$

In the above equation (2), only the value of secret message ($SM \in Z/I\mathbb{Z}$ with I prime) in ASCII is a_0 and the value of $a_1, \dots, a_{(h-1)}$ are at random in $Z/I\mathbb{Z}$. After that, we computed the $f(x)$ value which is converted into binary bit stream. Now, these secret bits are embedded into interpolated sub-sampled image block in different direction.

2.2 Directional pixel value ordering (DPVO)

In the PVO method, pixels within an image are ranked in ascending order form to obtain minimum and maximum pixel value. The hidden messages are stuffed into these

maximum or minimum pixel by changing its original value within an image block. When secret data embedding process occurs in different direction one after another (i. e. horizontal, vertical and diagonal) within 1st, 2nd, 3rd and so on minimum pixels for minimum modification based data embedding and in 1st, 2nd, 3rd and so on maximum pixels for maximum modification based data embedding, then this process is called DPVO. Following Fig. 3 depicted the basic diagram of DPVO.

After successful data embedding using DPVO, it is time to fetch the secret data from marked image block on the decoder side. In the data extraction procedure, it goes through reverse direction (i. e. diagonal, vertical and horizontal) of the embedding process.

2.3 Data embedding procedure

Secret data bit embedding process will begin in horizontally of the interpolated sub-sampled image. In our approach, the number of the modified pixel may be more than two which may affect in ascending order pixel ranking. So, a parameter α is subtracted from and add to 1st, 2nd, 3rd and so on minimum and maximum pixel value respectively to maintain same pixel ranking order. The α is only controlled by the original pixel block size. If a size of the original image block is $(w \times w)$ then, the minimal and maximal values of α will be 0 and $(w - 2)$ respectively.

Lemma 1 If the sub-sampled image size is $((2w - 1) \times (2w - 1))$ and the ranked pixel is $(p_1, p_2, \dots, p_{2w-2}, p_{2w-1})$, then, the adjusted value of pixel will be $(p_1 - \alpha_{(w-2)}, p_2 - \alpha_{(w-2)-1}, \dots, p_{w-1} - \alpha_{(w-2)-(w-2)}, p_w, p_{w+1} + \alpha_{((w-2)-(w-2))}, \dots, p_{2w-2} + \alpha_{((w-2)-1)}, p_{2w-1} + \alpha_{(w-2)})$, where, $\alpha_{((w-2)-(w-2))} = ((w - 2) - (w - 2))$. The minimal and maximal value of α will be 0 and $(w - 2)$ respectively.

Example 1 Assume that, the original block is $(w \times w) = (2 \times 2)$, then, interpolated sub-sampled image block will be (3×3) . The α 's maximum value $(w - 2) = 0$, which is added and subtracted to the maximum and from the minimum pixel value respectively. This process will continue until it (α) comes to zero. The Lemma 1's block diagram is presented in Fig. 4. $\square$

Fig. 3 Block diagram of Directional PVO

Fig. 4 Block diagram of Lemma 1

After that, the secret message will be placed according to the following procedure.

2.3.1 Minimum-modification-based data embedding

The pixels are taken from an original cover image. Sorting the pixels in row-wise of image block in increasing order to get $(x_{\sigma(1)}, \dots, x_{\sigma(n)})$, where $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ and $x_{\sigma(1)} \leq \dots \leq x_{\sigma(n)}$, $\sigma(j) > \sigma(i)$ if $x_{\sigma(j)} = x_{\sigma(i)}$ and $j > i$ is one to one mapping of n pixels $(x_1, \dots, x_n)$ of an image block X . We compute

$$d_{\min_k} = x_p - x_q \text{ where } \begin{cases} p = \min(\sigma((1) + k), \sigma((2) + k)), \\ q = \max(\sigma((1) + k), \sigma((2) + k)), \\ k = (0, 1, \dots, \text{fix}(n/2) - 1). \end{cases} \quad (3)$$

To round the elements towards zero, we use $\text{fix}()$ function which produces in an array of integers.

- $p = \sigma((1) + k)$ and $q = \sigma((2) + k)$ when $\sigma((1) + k) < \sigma((2) + k)$. Here, $d_{\min_k} \leq 0$.
- $p = \sigma((2) + k)$ and $q = \sigma((1) + k)$ when $\sigma((1) + k) > \sigma((2) + k)$. Here, $d_{\min_k} > 0$ and $x_{\sigma((1)+k)} < x_{\sigma((2)+k)}$.

Here, $C \in \{0, 1\}$ is a hidden data bit to embed within pixels. The value of α is changeable. The minimum pixel $x_{\sigma((1)+k)}$ is adjusted as x' by

$$x' = \begin{cases} (x_{\sigma((1)+k)} - \alpha) - C, & \text{if } d_{\min_k} = 0 \\ (x_{\sigma((1)+k)} - \alpha) - 1, & \text{if } d_{\min_k} < 0 \\ (x_{\sigma((1)+k)} - \alpha) - C, & \text{if } d_{\min_k} = 1 \\ (x_{\sigma((1)+k)} - \alpha) - 1, & \text{if } d_{\min_k} > 1 \end{cases} \quad (4)$$

Assume that, the marked pixel of X is $(y_1, y_2, \dots, y_n)$, where $y_{\sigma((1)+k)} = x'$ and also $y_i = x_i$ for all $i \neq \sigma((1) + k)$.

2.3.2 Maximum-modification-based data embedding

In this type of data embedding procedure, we compute

$$d_{\max_k} = x_r - x_s \text{ where } \begin{cases} r = \min(\sigma((n) - k), \sigma((n-1) - k)), \\ s = \max(\sigma((n) - k), \sigma((n-1) - k)), \\ k = (0, 1, \dots, \text{fix}(n/2) - 1). \end{cases} \quad (5)$$

- $r = \sigma((n-1) - k)$ and $s = \sigma((n) - k)$ when $\sigma((n) - k) > \sigma((n-1) - k)$. Here, $d_{\max_k} \leq 0$.
- $r = \sigma((n) - k)$ and $s = \sigma((n-1) - k)$ when $\sigma((n) - k) < \sigma((n-1) - k)$. Here, $d_{\max_k} > 0$ and $x_{\sigma((n)-k)} > x_{\sigma((n-1)-k)}$.

Here, $C \in \{0, 1\}$ is a hidden data bit to embed within pixel values. The value of α is changeable. The pixel(maximum) $x_{\sigma((n)-k)}$ is adjusted to x' by

$$x' = \begin{cases} (x_{\sigma((n)-k)} + \alpha) + C, & \text{if } d_{\max_k} = 0 \\ (x_{\sigma((n)-k)} + \alpha) + 1, & \text{if } d_{\max_k} < 0 \\ (x_{\sigma((n)-k)} + \alpha) + C, & \text{if } d_{\max_k} = 1 \\ (x_{\sigma((n)-k)} + \alpha) + 1, & \text{if } d_{\max_k} > 1 \end{cases} \quad (6)$$

Assume that, the marked pixel of X is $(y_1, y_2, \dots, y_n)$ where $y_{\sigma((n)-k)} = x'$ and also $y_i = x_i$ for all $i \neq \sigma((n) - k)$.

Now, the data embedding process in the horizontal direction is completed. After that, this procedure applied in the vertical direction and diagonal direction. Thus, construct a final marked image block. Figure 5 describes the overall block diagram of the data embedding process. "Algorithm 1" describes the proposed data embedding algorithm.

Algorithm 1: Data bit embedding process

Input: The original cover image ($CI_{(a \times b)}$). The hidden data SM .

Output: The interpolated sub-sampled stego images ESI'_1, ESI'_2, ESI'_3 and ESI'_4 .

Step-1: From original cover image create sub-sampled images using equation (1);

Step-2: From sub-sampled images construct interpolated sub-sampled images ESI_1, ESI_2, ESI_3 and ESI_4 using interpolation technique by average of neighbor pixels shown in Fig. 2;

Step-3: Convert secret message(a_0) into its ASCII value;

Step-4:

```

for  $x = 1$  to 4 do
    (1). Calculate the value of  $f(x)$  using  $f(x) = a_0 + a_{(h-2)}x^{(h-2)} + a_{(h-1)}x^{(h-1)}$ ;
    (2). Convert the value of  $f(x)$  into equivalent binary bit stream;
    (3). Take  $ESI_x$  interpolated sub-sampled image with size  $((a-1) \times (b-1))$ ;
    Sorting the pixel in ascending order in row wise;
    for  $row = 1$  to  $(a-1)$  do
        for pair of  $1^{st}$  to  $((a-1)/2)^{th}$  minimum and maximum pixel do
            |  $DataEmbed()$ ;
        end
    end
    Sorting the pixel in ascending order in column wise;
    for  $col = 1$  to  $(b-1)$  do
        for pair of  $1^{st}$  to  $((b-1)/2)^{th}$  minimum and maximum pixel do
            |  $DataEmbed()$ ;
        end
    end
    Sorting the pixel in ascending order in first diagonal;
    for  $diagonal1$  do
        for pair of  $1^{st}$  to  $((a-1)/2)^{th}$  minimum and maximum pixel do
            |  $DataEmbed()$ ;
        end
    end
    Sorting the pixel in ascending order in second diagonal;
    for  $diagonal2$  do
        for pair of  $1^{st}$  to  $((a-1)/2)^{th}$  minimum and maximum pixel do
            |  $DataEmbed()$ ;
        end
    end
    return  $ESI'_x$ ;
end

```

Step-5: Construct four interpolated sub-sampled stego images $ESI'_1, ESI'_2, ESI'_3, ESI'_4$;

Step-6 Stop;

Procedure: $DataEmbed()$

Calculate the $d_{min_k} = x_p - x_q$ and the $d_{max_k} = x_r - x_s$, where (p, q, r, s, k) are define in equation (3) and equation (5);

The marked value x' in minimum modification is computed using equation (4);

The marked value x' in maximum modification is computed using equation (6);

end Procedure

2.4 Numerical illustration of embedding process

Let us observe a numerical example of data bit embedding process to understand the proposed process. We take (4×4) original cover image. Create four sub-sampled images of size (2×2) as SI_1, SI_2, SI_3 and SI_4 from cover image. These four images are interpolated using interpolation techniques and make ESI_1, ESI_2, ESI_3 and ESI_4 of size (4×4) . Consider pixel values of sub-sampled image block (SI_1) is in first row (100, 102) and second row (107, 104). The first interpolate value is $\text{fix}((100 + 102)/2) = 101$ because of two neighbor pixel and for position (2, 2) there are four neighbor, so modified value will be $\text{fix}((100 + 102 + 107 + 104)/4) = 103$.

Let secret message is 'y' which we want to hide. We take ASCII value of message 'y' and then apply lagrange interpolation formula in equation (4) with $h = 3, l = 257, a_1 = 2, a_2 = 3$ in $Z / 257Z$. Convert the value of $f(x)$ in binary bit stream and embed these data bit into ESI_x where $x = 1, \dots, 4$, in different direction(horizontal, vertical, first and second diagonal) one after another. Thus makes four interpolated sub-sampled stego images ESI'_1, ESI'_2, ESI'_3 and ESI'_4 shows in Fig. 6.

2.5 Extraction process

In this phase, we implement data extraction process of the proposed scheme. As there are four shares, only 3 of them to reconstruct the message. So take anyone combination

Fig. 5 Block diagram of data embedding

of stego images (1, 2, 4), (2, 3, 4), (1, 3, 4) and (1, 2, 3). In each sub-sampled stego images among any one combination is used to perform data extraction process in the diagonal direction. We subtract and add the value of α from the maximum and to the minimum pixel of 1st, 2nd and so on respectively using the Lemma 2.

Lemma 2 If the interpolated sub-sampled stego image block is $((2w - 1) \times (2w - 1))$ and the ranked pixel is $(p'_1, p'_2, \dots, p'_{2w-2}, p'_{2w-1})$, then the adjusted pixel will be $(p'_1 + \alpha_{(w-2)}, p'_2 + \alpha_{((w-2)-1)}, \dots, p'_{w-1} + \alpha_{((w-2)-(w-2))}, p'_w, p'_{w+1} - \alpha_{((w-2)-(w-2))}, \dots, p'_{2w-2} - \alpha_{((w-2)-1)}, p'_{2w-1} - \alpha_{(w-2)})$, where the minimum value of $\alpha = 0$, the maximum value of $\alpha = (w - 2)$ and the $\alpha_{((w-2)-(w-2))} = ((w - 2) - (w - 2))$.

Example 2 Assume that, the interpolated sub-sampled stego image block size is (3×3) , then the sub-sampled image block size is (2×2) . The maximum value of α is 0, which is added and subtracted to the minimum and from the maximum pixel value respectively. The subtraction and addition process of α will be continue until it(α)

comes to 0. The Lemma 2's block diagram is presented in Fig. 7. $\square$

2.5.1 Minimum-modification-based data extraction

The data extraction process is occurs from the interpolated sub-sampled stego image block where the marked pixels are $(y_1, y_2, \dots, y_n)$. Here, pixel ranking remains same. We compute $d'_{\min_k} = y_p - y_q$, where (p, q, k) is described in equation (3).

* $y_p \leq y_q$ when $d'_{\min_k} \leq 0$. Now, $\sigma((1) + k) < \sigma((2) + k)$ and $p = \sigma((1) + k)$, $q = \sigma((2) + k)$:

- When $d'_{\min_k} \in \{0, -1\}$, there exist secret data and it is $C = -d'_{\min_k}$. The recovered pixel(minimum) is $x_{\sigma((1)+k)} = (y_p + \alpha) + C$;
- When $d'_{\min_k} < -1$, there absent secret data. The recovered minimum is $x_{\sigma((1)+k)} = (y_p + \alpha) + 1$.

Fig. 6 Numerical example of data embedding **Fig. 7** Block diagram of Lemma 2

* $y_p > y_q$ when $d'_{\min_k} > 0$. Now, $\sigma((1) + k) > \sigma((2) + k)$ and $p = \sigma((2) + k)$, $q = \sigma((1) + k)$:

- When $d'_{\min_k} \in \{1, 2\}$, there exist secret data and it is $C = d'_{\min_k} - 1$. The restored minimum is $x_{\sigma((1)+k)} = (y_q + \alpha) + C$;
- When $d'_{\min_k} > 2$, there absent secret data. The recovered pixel(minimum) is $x_{\sigma((1)+k)} = (y_q + \alpha) + 1$.

2.5.2 Maximum-modification-based data extraction

The image restoration procedure is conducted from the interpolated sub-sampled image where the marked value is $(y_1, y_2, \dots, y_n)$. Here, pixel ranking also remain the same.

Now, compute $d'_{\max_k} = y_k - y_l$ where (r, s, k) is described previously in equation (5).

* $y_r \leq y_s$ when $d'_{\max_k} \leq 0$. Now $\sigma((n-1) - k) < \sigma((n) - k)$ and $r = \sigma((n-1) - k)$, $s = \sigma((n) - k)$:

- When $d'_{\max_k} \in \{0, -1\}$, there exist secret data and it is $C = -d'_{\max_k}$. The recovered pixel(maximum) is $x_{\sigma((n)-k)} = (y_s - \alpha) - C$;
- When $d'_{\max_k} < -1$, there absent secret data. The recovered pixel(maximum) is $x_{\sigma((n)-k)} = (y_s - \alpha) - 1$.

* $y_r > y_s$ when $d'_{\max_k} > 0$. Now $\sigma((n-1) - k) > \sigma((n) - k)$ and $r = \sigma((n) - k)$, $s = \sigma((n-1) - k)$:

- When $d'_{\max_k} \in \{1, 2\}$, there exist secret data and it is $C = d'_{\max_k} - 1$. The pixel(maximum) is $x_{\sigma((n)-k)} = (y_r - \alpha) - C$;
- When $d'_{\max_k} > 2$, there absent secret data. The restored pixel(maximum) is $x_{\sigma((n)-k)} = (y_r - \alpha) - 1$.

The secret message, as well as the pixels, are recovered from two diagonal of interpolated sub-sampled stego image. The above-said process is applied through vertical and horizontal direction one after another. Finally, retrieve all the hidden data bit of corresponding $f(x)$. Now convert

the value of $f(x)$ into corresponding decimal value. Select anyone combination of stego images and apply Lagrange interpolation formula to recover secret data(a_0). After that, convert this ASCII value of secret (a_0) to its corresponding data. Now make sub-sampled images from interpolated sub-sampled images by eliminating all interpolate rows and columns. Finally, construct the real cover from sub-sampled images. Figure 8 shows the block diagram of the data extraction process. “Algorithm 2” describes the proposed data extraction algorithm.

Algorithm 2: Data bit extraction process

Input: The sub-sampled marked images ESI'_1, ESI'_2, ESI'_3 and ESI'_4 .

Output: Original cover image($CI_{(a \times b)}$). The secret message SM .

Step-1: To get back the hidden data, the receiver end can use any three sub-sampled stego images among four;

```

for count = 1 to 3 do
  for x = 1 to 4 do
    (1). Select  $ESI'_x$  interpolated sub-sampled stego image with size
       $((a-1) \times (b-1))$ ;
    Sorting the pixels in ascending order in second diagonal;
    for diagonal2 do
      for pair of  $((a-1)/2)^{th}$  to  $1^{st}$  maximum and minimum pixel do
        DataExtract();
      end
    end
    Sorting the pixel in ascending order in first diagonal;
    for diagonal1 do
      for pair of  $((a-1)/2)^{th}$  to  $1^{st}$  maximum and minimum pixel do
        DataExtract();
      end
    end
    Sorting the pixel in ascending order in column wise;
    for col =  $(b-1)$  to 1 do
      for pair of  $((a-1)/2)^{th}$  to  $1^{st}$  maximum and minimum pixel do
        DataExtract();
      end
    end
    Sorting the pixel in ascending order in row wise;
    for row =  $(a-1)$  to 1 do
      for pair of  $((a-1)/2)^{th}$  to  $1^{st}$  maximum and minimum pixel do
        DataExtract();
      end
    end
    return  $ESI_x$ , hidden bit( $C$ );
  end
end

```

end 2

. Convert these binary bit stream into equivalent decimal number;

Step-2: Select any one combination of x , (1,2,4), (1,3,4), (2,3,4) and (1,2,3). Then apply lagrange interpolation polynomial equation to recover the secret data(a_0) which are extracted from any three interpolated sub-sampled stego images;

Step-3: Convert this ASCII value of (a_0) to its corresponding data;

Step-4: Construct sub-sampled images from interpolated sub-sampled images;

Step-5: From sub-sampled images construct the original cover image($CI_{(a \times b)}$);

Step-6 Stop;

Procedure: *DataExtract()*

Calculate the $d'_{\max_k} = y_r - y_s$ and the $d'_{\min_k} = y_p - y_q$, where (r, s, p, q, k) are defined in equation (5) and equation (3);

The maximum pixel value is restored through equation (6);

The minimum pixel value is recovered using equation (4);

end procedure

Fig. 8 Block diagram of data extraction

2.6 Numerical illustration of extraction process

Let us observe an example of data extraction to recognize the proposed method. We take four interpolated sub-sampled stego images ESI'_1 , ESI'_2 , ESI'_3 and ESI'_4 of size (3×3) shown in Fig. 9.

The data extraction process goes through diagonal, vertical and then horizontal direction one after another of each interpolated stego image of above-said any one combination. Retrieve all the pixel value and hidden data of stego images. Then convert these binary data bit to its equivalent decimal value. Apply Lagrange interpolation equation to reconstruct the hidden data (a_0) from any three stego image. Finally, convert this ASCII value of (a_0) to its corresponding data. Afterward, the original cover image is built from sub-sampled images shown in Fig. 10.

3 Experimental results and comparisons

We use standard cover images (gray-scale) from different databases. Ten images (512×512) were taken from USC-SIPI [30], twenty images (481×321) were taken from Berkeley Segmentation Dataset and Benchmark [31], twenty-four images (768×512) or (512×768) were taken from Kodak [9] and twenty images with different size were taken from the National Library of Medicine [26] for our experiment as test images and only 10 of each data set are shown in Fig. 11. For analysis the usefulness of our proposed method, we are used four different benchmark image databases. The secret message embedding and extraction algorithm for the proposed scheme are examined through MATLAB R2014a (8.3.0.532). The capacity of the data bit to an image with quality measured with PSNR (dB). For embedding

Fig. 9 Numerical example of data extraction **Fig. 10** Recover of cover image

Fig. 11 The standard cover images are used in this scheme

6,42,168 secret bits, the PSNR result shows 45.1081 dB for Lena image presented in Table 1. Tables 2, 3 and 4 are displayed the PSNR values with the payload of 30,000 bits from other datasets. These table presented that PSNR value is greater than 55 dB after embedding 30,000 secret bits. The proposed method improves the value of PSNR (dB) and payload than the other data hiding method with PVO described by [13, 19, 22, 24, 25]. The comparison results in terms of PSNR(dB) between the proposed and the other data hiding PVO methods with the embedding capacity of 10,000 bits are appeared in Table 5.

The data embedding capacity is dependent on image size which is displayed in Fig. 12. It is also shows the better

performance for image block size (3 × 3) with good PSNR value than the other block size. Using (2 × 2) block size in Lena image (512 × 512), 4,46,196 bits can be embedded with PSNR value 41.58 dB whereas 6,42,168 bits are embedded with PSNR value 45.10 dB with the block size (3 × 3).

The performance comparison graph of PSNR (dB) with different schemes is displayed in Fig. 13. The proposed method enhances the payload compared with other PVO methods. For Lena image, the payload is 610168, 604168, 605168, 596168, 512168 and 523182 bits more than Li et al.'s [13], Peng et al.'s [24], Ou et al.'s [22], Qu et al.'s [25], Jana [6] and Meikap et al.'s [19] method respectively depicted in Table 6. The differentiation graph including

Table 1 The data bit embedding capacity (EC) with different pay load (bits) of USC-SIPI 10 standard test images with PSNR (dB)

Cover image (CI)	Data (bits)	PSNR (ESI_1 vs ESI'_1)	PSNR (ESI_2 vs ESI'_2)	PSNR (ESI_3 vs ESI'_3)	PSNR (ESI_4 vs ESI'_4)
Lena	30,000	60.0308	60.0224	60.9691	60.0095
	100,000	54.2329	54.2033	54.1779	54.1641
	300,000	48.9261	48.9001	48.9143	48.9160
	6,42,168	45.1081	45.1013	45.0997	45.0943
Airplane F16	30,000	59.4728	59.5083	59.4351	59.5142
	100,000	54.5837	54.6013	54.5579	54.6090
	300,000	50.0480	50.0639	50.0458	50.0666
	7,65,010	45.4585	45.4651	45.4617	45.4513
Baboon	30,000	55.8165	55.7900	55.7392	55.6767
	100,000	50.7001	50.6867	50.6597	50.7102
	300,000	47.1044	47.1118	47.0013	47.0986
	5,40,245	44.8035	44.8013	44.8035	44.8020
Elaine	30,000	59.1069	59.1972	58.9189	58.9341
	100,000	53.8720	53.8683	53.8965	53.8905
	300,000	48.1806	48.7869	48.8204	47.9237
	6,31,689	45.0599	45.0510	45.0603	45.0534
Fishing boat	30,000	59.9766	59.0464	60.0011	60.0102
	100,000	54.6054	54.1181	54.0575	53.9906
	300,000	49.5202	49.0073	48.9970	49.1986
	6,96,635	45.2234	45.2214	45.2329	45.2209
House	30,000	60.6249	60.2861	60.4119	61.0963
	100,000	55.2581	55.6210	55.6402	55.4141
	300,000	49.7354	49.9788	49.9166	49.9583
	7,35,202	45.3607	45.3639	45.3634	45.3604
Peppers	30,000	58.4467	58.9680	58.0028	58.1158
	100,000	53.4042	53.5771	53.6030	53.5919
	300,000	48.5372	48.0889	48.1553	48.1973
	6,40,845	45.1068	45.1091	45.1083	45.1197
Sailboat on lake	30,000	57.5059	57.7179	57.7680	57.7687
	100,000	52.7810	52.2517	52.3077	53.0320
	300,000	47.8125	47.7759	47.9727	47.9741
	5,68,674	44.8651	44.8599	44.9638	44.8621
Splash	30,000	57.7015	57.8507	58.0165	57.8975
	100,000	52.7727	52.3167	51.9972	52.3128
	300,000	48.3853	48.4383	48.2812	48.3530
	6,74,973	45.2939	45.3124	45.3927	45.3963
Tiffany	30,000	60.2009	60.1691	60.1694	60.1843
	100,000	54.8347	54.8379	54.8300	54.8419
	300,000	50.0615	50.0754	50.0712	50.0884
	8,39,463	45.6250	45.6300	45.6355	45.6292

bit capacity with various PVO based schemes is shown in Fig. 14. It is noticeable that in the purpose of the payload as well as quality, our method is superior to previous PVO methods.

3.1 Steganographic attacks

Steganalysis technique is a valuable activity in hidden message communication where a suspected image has

Table 2 Compute the PSNR (dB) for the Kodak images with the payload 30,000 bits

Cover image (CI)	PSNR (ESI_1 vs ESI'_1)	PSNR (ESI_2 vs ESI'_2)	PSNR (ESI_3 vs ESI'_3)	PSNR (ESI_4 vs ESI'_4)
kodim01	59.4209	59.4403	59.0846	59.0544
kodim02	58.5933	58.7028	58.1770	58.1733
kodim03	62.2934	62.4061	61.9761	61.8781
kodim04	61.4075	61.3699	61.1351	61.1903
kodim05	59.7293	59.6866	58.8724	58.7862
kodim06	61.9675	61.9869	61.1067	61.1065
kodim07	61.2331	61.2372	61.0314	61.1236
kodim08	60.0424	60.0359	59.9328	59.9557
kodim09	61.5466	61.3777	61.0202	61.1571
kodim10	61.1422	61.1179	60.8224	60.8478
kodim11	62.0161	62.0698	62.3510	62.3442
kodim12	61.2300	61.1838	61.2361	61.1935
kodim13	60.2932	60.2849	60.2638	60.2576
kodim14	58.0893	58.0547	57.7902	57.8439
kodim15	60.2173	60.1860	60.1543	60.1519
kodim16	63.1452	63.1632	62.2844	62.1909
kodim17	61.8148	61.7890	62.0275	62.0789
kodim18	57.1546	57.2518	57.1733	57.2758
kodim19	60.7253	60.6613	60.6020	60.6528
kodim20	65.1076	65.1481	65.5565	65.5602
kodim21	61.3161	61.4953	60.6027	60.5791
kodim22	61.6053	61.7036	61.6522	61.6385
kodim23	62.4947	62.6207	62.5790	62.5175
kodim24	62.8106	62.8101	62.7832	62.6986

Table 3 Compute the PSNR (dB) for the images of Berkeley Segmentation Dataset with the payload 30,000 bits

Cover image (CI)	PSNR (ESI_1 vs ESI'_1)	PSNR (ESI_2 vs ESI'_2)	PSNR (ESI_3 vs ESI'_3)	PSNR (ESI_4 vs ESI'_4)
3096	58.6916	58.6497	58.5925	58.4657
8023	56.2949	56.3324	56.3137	56.2708
21077	61.0109	60.8583	60.8933	61.0014
35058	56.7286	56.7037	56.7252	56.6267
35091	60.0031	59.2180	59.2037	59.1979
41069	59.8362	59.8686	59.8845	59.8292
69020	57.3592	57.3699	57.3876	57.3295
69040	56.0710	56.0636	56.0857	56.0452
85048	58.9009	59.0262	58.9468	58.9885
87046	59.0203	59.7342	58.9968	58.8593
105025	57.7594	57.8242	57.8054	57.8496
109034	55.3072	56.3124	55.1947	55.3374
109053	58.5136	59.2002	59.0011	58.9970
113044	57.8196	57.8137	57.9879	58.0170
157055	59.4276	59.2559	59.4356	59.3173
160068	57.9579	58.0623	58.0478	57.6848
163085	59.3265	58.9918	59.2908	59.2343
236037	59.2985	59.2908	59.2590	59.3120
253055	58.3930	57.9051	58.1994	58.4380
299086	58.8721	59.0365	58.8628	58.1069

Table 4 Compute the PSNR (dB) for the image of National Library of Medicine Dataset with the payload 30,000 bits

Cover image (CI)	PSNR (ESI_1 vs ESI'_1)	PSNR (ESI_2 vs ESI'_2)	PSNR (ESI_3 vs ESI'_3)	PSNR (ESI_4 vs ESI'_4)
CXR1000_IM-0003-1001	61.5486	61.5690	61.5570	61.5570
CXR1000_IM-0003-3001	60.5041	60.4758	61.5467	60.5469
CXR1025_IM-0020-1001	59.4676	58.9183	58.5996	59.0134
CXR1037_IM-0029-1001	60.0501	60.9341	61.2166	60.1045
CXR1074_IM-0054-1001	59.2476	59.5074	60.0428	59.3276
CXR1109_IM-0076-1001	61.3409	60.9678	61.1367	60.9134
CXR1114_IM-0079-1001	59.0756	59.0322	58.9349	58.9609
CXR1119_IM-0080-1001	58.3439	58.3369	58.1208	58.2302
CXR1120_IM-0080-1001	56.7626	57.6120	56.7328	57.6761
CXR1151_IM-0102-1001	59.1244	59.1164	58.7986	59.0914
CXR1188_IM-0127-1001	61.9989	62.2274	62.2311	62.1107
MPX1355_synpic55464	60.6871	60.7803	60.2471	60.7141
MPX1430_synpic60423	59.3604	59.3780	60.0071	60.0142
MPX1459_synpic43454	60.1727	59.8446	60.2539	59.9883
MPX1521_synpic52763	57.6684	57.9687	58.2519	58.3073
MPX1726_synpic16586	56.6520	56.7505	56.7132	56.6637
MPX1726_synpic16587	58.3329	58.4372	58.4823	58.4913
MPX1855_synpic23323	61.1139	61.1915	61.1285	61.2235
MPX2720_synpic3995	62.8506	62.6011	62.7091	62.6497
MPX2750_synpic3136	57.2539	57.2317	57.3276	57.3324

Table 5 Comparison between the other schemes (Li et al. [13], Peng et al. [24], Ou et al. [22] and Qu et al. [25] and Meikap et al. [19]) and the proposed scheme of PSNR (dB) with bit capacity of 10,000 bits

Cover image (CI)	Li et al. [13]	Peng et al. [24]	Ou et al. [22]	Qu et al. [25]	Meikap et al. [19]	Proposed
Airplane F 16	61.6	62.9	63.3	63.7	63.1	64.2
Baboon	53.5	53.5	54.5	54.2	55.0	60.6
Barbara	59.9	60.5	60.6	59.8	60.4	61.6
Elaine	56.8	57.3	57.4	58.7	58.1	64.1
Fishing boat	57.8	58.2	58.1	58.4	59.2	64.9
House	61.8	64.4	63.7	64.6	66.4	65.0
Lena	59.8	60.4	60.6	60.3	59.8	65.6
Peppers	58.5	58.9	59.2	58.8	57.8	62.7
Sailboat on lake	58.2	58.8	59.4	59.8	59.1	61.9
Tiffany	60.1	60.7	60.3	60.6	59.7	64.9
Average	58.8	59.5	59.7	59.8	59.9	63.6

Fig. 12 The embedding performance of proposed method with different block sizes (for Lena image)

hidden data or not. Nowadays, these systems do not fulfill a sufficient security. So, users leave hints while data embedding into an image. Therefore, a steganalyst identifies whether a hidden message exists or not in an image. All the steganalyst performs this in different ways. This way is categorized in two part: Targeted and Blind steganalysis. Among the targeted method, the structural attack, statistical attack and visual attack are there. On the other hand, one important method of blind steganalysis is Regular Singular (RS) analysis proposed by J. Fridrich [4].

3.2 RS analysis

We examine sub-sampled marked images across the RS analysis method [4]. If the RS value is nearer to zero, then

Fig. 13 Comparisons of PSNR (dB) between the schemes of Li et al. [13], Peng et al. [24], Ou et al. [22], Qu et al. [25], Meikap et al. [19] and proposed method

Table 6 Comparison of payload in number of bits

Cover image (CI)	Li et al. [13]	Peng et al. [24]	Ou et al. [22]	Qu et al. [25]	Jana [6]	Meikap et al. [19]	Proposed
Airplane F 16	38,000	52,000	47,000	69,000	-	1,70,334	7,65,010
Baboon	13,000	13,000	13,000	15,000	-	1,44,767	5,40,245
Barbara	27,000	29,000	31,000	33,000	1,30,000	1,30,056	6,17,418
Elaine	21,000	24,000	23,000	29,000	-	81,302	6,31,689
Fishing boat	24,000	26,000	26,000	30,000	-	80,504	6,96,635
House	30,000	46,000	37,000	64,000	-	1,55,622	7,35,202
Lena	32,000	38,000	37,000	46,000	1,30,000	1,18,986	6,42,168
Peppers	28,000	30,000	31,000	33,000	1,30,000	95,312	6,40,845
Sailboat on lake	23,000	26,000	26,000	29,000	-	80,974	5,68,674
Tiffany	33,000	40,000	40,000	52,000	1,30,000	1,39,618	8,39,463
Average	26,900	32,400	31,100	40,000	1,30,000	1,19,747	6,67,734

slighter changes occurred and it is safe. Table 7 described the R_M and R_{-M} , S_M and S_{-M} values are closely equal. Both $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ rule satisfies for the marked image blocks. So, our proposed method is more secure for RS attack.

3.3 Statistical attack

We examine the strength of our proposed method by concealing the hidden data. Our method is safe to prevent the various attacks due to its having no adverse effect. The correlation coefficient(CC) between the original cover image and the marked image is shown in Table 8. It displays, the CC is close to one which represents the best privacy for data hiding. The standard deviation(SD) between the original and marked image is depicted in Table 9. It displays, the SD is close to zero which represents the best privacy for data hiding. Our proposed method obtained the strength against various attacks. Without loss of any data, the hidden message and cover image recovered from sub-sampled marked (stego) images.

3.4 Structural similarity index and normalized cross-correlation

The structural similarity(SSIM) index is a technique to determine the similarity between the original and the marked image. It is based on three computation key: the luminance key, the structural key the contrast key. The whole index is a multiplicative combination of these three keys. In Normalised cross-correlation (NCC), we find out some common patterns between cover and stego image. The NCC has been used in the fields of block matching, image and video compression and image registration etc. Table 10 shows the value of SSIM and NCC between original and stego image.

4 Conclusion

A new RDH approach for secret sharing within the sub-sampled image using Directional Pixel Value Ordering (DPVO) is proposed. The sub-sampled images from the cover image help to carry secret information in a

Fig. 14 Comparisons among the other schemes (Li et al. [13], Peng et al. [24], Ou et al. [22], Qu et al. [25] and Meikap et al. [19]) and the proposed scheme of payload

Table 7 RS analysis for stego image

Image Database	Cover Image (CI)	Stego (Image)	R_M	R_{-M}	S_M	S_{-M}	RS value
USC-SIPI (512 × 512)	Lena	ESI'_1	34402	32843	30495	32054	0.0480
		ESI'_2	34357	32887	30540	32010	0.0453
		ESI'_3	34528	32763	30369	32134	0.0544
		ESI'_4	34461	32814	30436	32083	0.0508
	Airplane F16	ESI'_1	34124	31455	30773	33442	0.0823
		ESI'_2	34228	31353	30669	33544	0.0886
		ESI'_3	34178	31442	30719	33455	0.0843
		ESI'_4	34219	31360	30678	33537	0.0881
BSDS (481 × 321)	69040	ESI'_1	19724	19952	18237	18009	0.0120
		ESI'_2	19673	19960	18288	18001	0.0151
		ESI'_3	19765	19913	18196	18048	0.0078
		ESI'_4	19723	19875	18238	18086	0.0080
	87046	ESI'_1	19976	19389	17985	18572	0.0309
		ESI'_2	20261	19189	17700	18772	0.0565
		ESI'_3	20070	19367	17891	18594	0.0370
		ESI'_4	20055	19331	17906	18630	0.0381
Kodak (512 × 768)	kodim18	ESI'_1	52616	52593	44793	44816	0.0005
		ESI'_2	52576	52825	44833	44584	0.0051
		ESI'_3	52329	52628	45080	44781	0.0061
		ESI'_4	52720	52427	44689	44982	0.0060
	kodim19	ESI'_1	52274	50246	45135	47163	0.0416
		ESI'_2	51801	50798	45608	46611	0.0206
		ESI'_3	51950	50610	45459	46799	0.0275
		ESI'_4	52019	50571	45390	46838	0.0297
National Library of Medicine (512 × 420)	CXRI109_IM-0076-1001	ESI'_1	29122	27226	24091	25987	0.0713
		ESI'_2	29006	27287	24207	25926	0.0646
		ESI'_3	28953	27382	24260	25831	0.0590
		ESI'_4	29010	27264	24203	25949	0.0656
	CXRI114_IM-0079-1001	ESI'_1	29564	29031	23649	24182	0.0200
		ESI'_2	29470	28964	23743	24249	0.0190
		ESI'_3	29710	28965	23503	24248	0.0280
		ESI'_4	29562	28951	23651	24262	0.0230

distributed manner. The proposed scheme improves the level of security due to Lagrange function and hard to reveal by the adversary. The new algorithms for embedding and extraction are designed in such a manner that more data bits are embedded and extracted to and from

interleaved pixels respectively. Our method accomplishes secure data communication because we embed the Lagrange function generated data only, not the real hidden message. The hidden message obtains through-out a predefined threshold number of the sub-sampled

Table 8 The correlation coefficient(CC)between the original and marked image

Image database	Cover image (CI)	Data (bits)	CC			
			$ESI_1 \& ESI'_1$	$ESI_2 \& ESI'_2$	$ESI_3 \& ESI'_3$	$ESI_4 \& ESI'_4$
USC-SIPI (512 × 512)	Lena	30,000	1.0000	1.0000	1.0000	1.0000
		100,000	0.9999	0.9999	0.9999	0.9999
		300,000	0.9998	0.9998	0.9998	0.9998
		6,42,168	0.9995	0.9995	0.9995	0.9995
	Airplane F16	30,000	1.0000	1.0000	1.0000	1.0000
		100,000	1.0000	1.0000	1.0000	1.0000
		300,000	0.9999	0.9999	0.9999	0.9999
		7,65,010	0.9998	0.9998	0.9998	0.9998
BSDS (481 × 321)	69040	30,000	0.9998	0.9998	0.9998	0.9998
		100,000	0.9993	0.9993	0.9993	0.9993
		199,617	0.9987	0.9987	0.9987	0.9987
	87046	30,000	0.9999	0.9999	0.9999	0.9999
		100,000	0.9998	0.9998	0.9998	0.9998
		300,000	0.9996	0.9996	0.9996	0.9996
Kodak (512 × 768)	kodim18	3,59,889	0.9996	0.9996	0.9996	0.9996
		30,000	0.9999	0.9999	0.9999	0.9999
		100,000	0.9998	0.9998	0.9998	0.9998
		300,000	0.9994	0.9994	0.9994	0.9994
	kodim19	6,70,304	0.9989	0.9989	0.9989	0.9989
		30,000	1.0000	1.0000	1.0000	1.0000
		100,000	0.9999	0.9999	0.9999	0.9999
		300,000	0.9998	0.9998	0.9998	0.9998
National Library of Medicine (512 × 420)	CXR110 9_IM-00 76-1001	9,01,696	0.9994	0.9994	0.9994	0.9994
		30,000	1.0000	1.0000	1.0000	1.0000
		100,000	0.9999	0.9999	0.9999	0.9999
		300,000	0.9996	0.9996	0.9996	0.9996
	CXR111 4_IM-00 79-1001	6,64,532	0.9994	0.9994	0.9994	0.9994
		30,000	1.0000	1.0000	1.0000	1.0000
		100,000	0.9998	0.9998	0.9998	0.9998
		300,000	0.9996	0.9996	0.9996	0.9996
		6,50,888	0.9992	0.9992	0.9992	0.9992

image, not the entire sub-sampled image. Using method, we obtained an average PSNR value over 60 dB and average data embedding capacity over 6,00,000 bits. The proposed method is examined by RS analysis, SD, CC, SSIM

and NCC that gives good results. We also notice that this scheme shows an excellent performance than other PVO based works and is secure against several steganographic attacks.

Table 9 The standard deviation (SD) between the original and marked image

Image database	Cover image	Data (bits)	SD											
			ESI_1	ESI'_1	Difference	ESI_2	ESI'_2	Difference	ESI_3	ESI'_3	Difference	ESI_4	ESI'_4	Difference
USC-SIP1 (512 × 512)	Lena	30,000	39.1769	39.1985	0.0216	39.1842	39.2059	0.0217	39.1847	39.2063	0.0217	39.1872	39.2087	0.0215
		100,000		39.2545	0.0776		39.2626	0.0784		39.2630	0.0783		39.2656	0.0784
		300,000		39.4549	0.2780		39.4628	0.2786		39.4622	0.2775		39.4646	0.2774
		6,42,168		39.7881	0.6112		39.7966	0.6124		39.7965	0.6118		39.7994	0.6122
BSDS (481 × 321)	Airplane F16	30,000	48.4544	48.4851	0.0307	48.4892	48.5194	0.0302	48.4345	48.4649	0.0304	48.4694	48.4993	0.0300
		100,000		48.5589	0.1045		48.5931	0.1038		48.5389	0.1044		48.5732	0.1038
		300,000		48.7699	0.3155		48.8027	0.3135		48.7497	0.3152		48.7833	0.3139
		7,65,010		49.2497	0.7953		49.2832	0.7940		49.2287	0.7942		49.2631	0.7937
BSDS (481 × 321)	69040	30,000	25.7045	25.7514	0.0469	25.7308	25.7773	0.0465	25.7074	25.7539	0.0465	25.7153	25.7610	0.0457
		100,000		25.9376	0.2331		25.9583	0.2274		25.9394	0.2319		25.9463	0.2309
		199,617		26.4275	0.7230		26.4563	0.7255		26.4319	0.7245		26.4398	0.7244
		87046	41.2540	41.2884	0.0344	41.2508	41.2855	0.0347	41.3117	41.3460	0.0343	41.3051	41.3399	0.0348
BSDS (481 × 321)	87046	100,000		41.3888	0.1348		41.3856	0.1348		41.4464	0.1347		41.4410	0.1359
		300,000		41.8094	0.5554		41.8068	0.5560		41.8688	0.5572		41.8625	0.5574
		3,59,889		41.9537	0.6997		41.9514	0.7006		42.0129	0.7012		42.0072	0.7021

Table 10 The data embedding capacity (EC) with different pay load (data bits) of four different image databases with structural similarity(SSIM) index and normalized cross-correlation (NCC)

Image database	Cover image (CI)	Data (bits)	ESI_1 vs ESI'_1		ESI_2 vs ESI'_2		ESI_3 vs ESI'_3		ESI_4 vs ESI'_4	
			SSIM	NCC	SSIM	NCC	SSIM	NCC	SSIM	NCC
USC-SIPI (512 × 512)	Lena	30,000	1.0000	1.0000	1.0000	1.0000	0.9999	1.0000	0.9999	1.0000
		100,000	0.9997	0.9999	0.9997	0.9999	0.9997	0.9999	0.9997	0.9999
		300,000	0.9987	0.9998	0.9987	0.9998	0.9987	0.9998	0.9986	0.9998
		6,42,168	0.9961	0.9995	0.9961	0.9995	0.9961	0.9995	0.9961	0.9995
	Airplane F16	30,000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
		100,000	0.9999	1.0000	0.9999	1.0000	0.9999	1.0000	0.9999	1.0000
		300,000	0.9997	0.9999	0.9997	0.9999	0.9997	0.9999	0.9997	0.9999
		7,65,010	0.9990	0.9998	0.9990	0.9998	0.9990	0.9998	0.9990	0.9998
BSDS (481 × 321)	69040	30,000	0.9989	0.9998	0.9990	0.9998	0.9989	0.9998	0.9990	0.9998
		100,000	0.9968	0.9993	0.9969	0.9993	0.9969	0.9993	0.9969	0.9993
		199,617	0.9944	0.9987	0.9944	0.9987	0.9944	0.9987	0.9944	0.9987
		87046	30,000	0.9995	0.9999	0.9995	0.9999	0.9996	0.9999	0.9995
	87046	100,000	0.9991	0.9998	0.9991	0.9998	0.9991	0.9998	0.9991	0.9998
		300,000	0.9978	0.9996	0.9978	0.9996	0.9978	0.9996	0.9978	0.9996
		3,59,889	0.9976	0.9996	0.9976	0.9996	0.9976	0.9996	0.9976	0.9996

Compliance with ethical standards

Conflict of interest The authors declare that they have no competing interests.

References

- Alattar AM (2004) Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans Image Process* 13(8):1147–1156
- Chang YT, Huang CT, Lee CF, Wang SJ (2013) Image interpolating based data hiding in conjunction with pixel-shifting of histogram. *J Supercomput* 66(2):1093–1110
- Chowdhuri P, Pal P, Jana B (2018) A new dual image-based steganographic scheme for authentication and tampered detection. *Inf Technol Appl Math ICITAM 2017* 699:163
- Fridrich J, Goljan M, Du R (2001) Reliable detection of LSB steganography in color and grayscale images. In: *Proceedings of the 2001 workshop on Multimedia and security: new challenges*, ACM, 27–30
- Jana B (2016) Dual image based reversible data hiding scheme using weighted matrix. *Int J Electron Inf Eng* 5(1):6–19
- Jana B (2017) Reversible data hiding scheme using sub-sampled image exploiting Lagrange's interpolating polynomial. *Multimed Tools Appl* 77:1–17
- Jana B, Giri D, Mondal SK (2018) Dual image based reversible data hiding scheme using (7, 4) hamming code. *Multimed Tools Appl* 77(1):763–785
- Kim KS, Lee MJ, Lee HY, Lee HK (2009) Reversible data hiding exploiting spatial correlation between sub-sampled images. *Pattern Recognit* 42(11):3083–3096
- Kodak lossless true color image suite, <http://r0k.us/graphics/kodak/>
- Lee CF, Huang YL (2012) An efficient image interpolation increasing payload in reversible data hiding. *Expert Syst Appl* 39(8):6712–6719
- Lee CF, Chang CC, Gao CY (2013) A two-staged multi-level reversible data hiding exploiting lagrange interpolation. In: *2013 Ninth international conference on intelligent information hiding and multimedia signal processing, IEEE*, (pp. 485–488)
- Lee CF, Weng CY, Chen KC (2017) An efficient reversible data hiding with reduplicated exploiting modification direction using image interpolation and edge detection. *Multimed Tools Appl* 76(7):9993–10016
- Li X, Li J, Li B, Yang B (2013) High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Process* 93(1):198–205
- Lin CC, Tai WL, Chang CC (2008) Multilevel reversible data hiding based on histogram modification of difference images. *Pattern Recognit* 41(12):3582–3591
- Lu TC, Chang CC, Huang YH (2014) High capacity reversible hiding scheme based on interpolation, difference expansion, and histogram shifting. *Multimed Tools Appl* 72(1):417–435
- Luo L, Chen Z, Chen M, Zeng X, Xiong Z (2010) Reversible image watermarking using interpolation technique. *IEEE Trans Inf Forensics Secur* 5(1):187–193
- Luo H, Yu FX, Chen H, Huang ZL, Li H, Wang PH (2011) Reversible data hiding based on block median preservation. *Inf Sci* 181(2):308–328
- Malik A, Sikka G, Verma HK (2017) An image interpolation based reversible data hiding scheme using pixel value adjusting feature. *Multimed Tools Appl* 76(11):13025–13046
- Meikap S, Jana B (2018) Directional PVO for reversible data hiding scheme with image interpolation. *Multimed Tools Appl* 77(23):31281–31311
- Ni Z, Shi YQ, Ansari N, Su W (2006) Reversible data hiding. *IEEE Trans Circuits Syst Video Technol* 16(3):354–362
- Ou B, Li X, Wang J (2016) Improved PVO-based reversible data hiding: a new implementation based on multiple histograms modification. *J Vis Commun Image Represent* 38:328–339
- Ou B, Li X, Zhao Y, Ni R (2014) Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. *Signal Process Image Commun* 29(7):760–772

23. Pal P, Chowdhuri P, Jana B (2018) Weighted matrix based reversible watermarking scheme using color image. *Multimed Tools Appl*, 1–26
24. Peng F, Li X, Yang B (2014) Improved PVO-based reversible data hiding. *Digit Signal Process* 25:255–265
25. Qu X, Kim HJ (2015) Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. *Signal Process* 111:249–260
26. The National Library of Medicine presents MedPix®, <https://openi.nlm.nih.gov/gridquery.php?q=&it=x>
27. Tian J (2003) Reversible data embedding using a difference expansion. *IEEE Trans Circuits Syst Video Technol* 13(8):890–896
28. Tsai P, Hu YC, Yeh HL (2009) Reversible image hiding scheme using predictive coding and histogram shifting. *Signal Process* 89(6):1129–1143
29. Tsai Y.Y, Chen J.T, Kuo Y.C, Chan C.S (2014) A generalized image interpolation-based reversible data hiding scheme with high embedding capacity and image quality. *KSII Trans Internet Inf Syst*, 8(9)
30. University of Southern California, The USC-SIPI image database, <http://sipi.usc.edu/database/database.php?volume=misc>
31. University of California, Berkeley, The Berkeley segmentation dataset and benchmark, <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300-images.tgz>
32. Wahed M.A, Nyeem H (2017, September) A simplified parabolic interpolation based reversible data hiding scheme. In: 2017 4th International conference on advances in electrical engineering (ICAEE), IEEE, (pp. 743–748)
33. Wahed M.A, Nyeem H (2017, December) Efficient Data Embedding for Interpolation based Reversible Data Hiding Scheme. In: 2017 2nd international conference on electrical and electronic engineering (ICEEE), IEEE, (pp. 1–4)
34. Wahed M.A, Nyeem H (2018) Reversible data hiding with interpolation and adaptive embedding. *Multimed Tools Appl*, 1–25
35. Wahed MA, Nyeem H (2019) High capacity reversible data hiding with interpolation and adaptive embedding. *PloS one* 14(3):e0212093
36. Wang XT, Chang CC, Nguyen TS, Li MC (2013) Reversible data hiding for high quality images exploiting interpolation and direction order mechanism. *Digit Signal Process* 23(2):569–577
37. Zhang X, Sun Z, Tang Z, Yu C, Wang X (2017) High capacity data hiding based on interpolated image. *Multimed Tools Appl* 76(7):9195–9218

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.