Arabian Journal for Science and Engineering (2019) 44:2995–3004
https://doi.org/10.1007/s13369-018-3372-2

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Very High Capacity Image Steganography Technique Using Quotient Value Differencing and LSB Substitution

Gandharba Swain[1]

Received: 12 February 2018 / Accepted: 31 May 2018 / Published online: 16 June 2018

Abstract
This article proposes a very high capacity steganography technique using differencing and substitution mechanisms. It divides the image into non-overlapped 3×3 pixel blocks. For every pixel of a block, least significant bit (LSB) substitution is applied on two LSBs and quotient value differencing (QVD) is applied on the remaining six bits. Thus, there are two levels of embedding: (i) LSB substitution at lower bit planes and (ii) QVD at higher bit planes. If a block after embedding in this fashion suffers with fall off boundary problem, then that block is undone from the above hybrid embedding and modified 4-bit LSB substitution is applied. Experimentally, it is evidenced that the hiding capacity is improved to a greater extent. It is also experimentally proved that pixel difference histogram and RS analysis techniques cannot detect the proposed steganography technique.

Keywords Steganography · LSB substitution · QVD · RS analysis · PDH analysis

1 Introduction

Steganography is a technique of secret communication. It is performed by injecting the secret data inside image, audio, and video files [1]. In image steganography, the statistical properties of the image should be preserved after hiding the secret data in it [2]. Least significant bit (LSB) substitution and pixel value differencing (PVD) are the most popular steganography techniques in spatial domain. But RS analysis [3] can detect the LSB substitution techniques, and pixel difference histogram (PDH) analysis can detect the PVD techniques [5]. The PVD techniques hide lesser number of bits in smooth regions and more number of bits in edge regions. The first PVD steganography technique proposed by Wu and Tsai [4] uses pixel value differencing in 1×2 pixel blocks, wherein the hiding capacity is very less. The hiding capacity of PVD steganography is improved by using 2×2 pixel blocks [6,7]. Wu et al. [8] combined the LSB substitution and PVD to achieve better performance. They applied 3-bit LSB substitution in smooth regions and PVD steganography in edge regions. Yang et al. [9] found that Wu et al.'s scheme uses LSB substitution in most of the blocks, so it is likely to be detected by RS analysis. A mixture of LSB substitution and PVD based on addition and subtraction mechanism was proposed using 1×3 pixel blocks [10]. Data are embedded in the middle pixel by LSB substitution, and then the two neighboring pixels on left and right are embedded using PVD. This technique suffers with fall off boundary problem (FOBP) and attacked by PDH analysis. Swain achieved higher performance by extending the Khodaei and Faez's idea to 2×2 pixel blocks [11]. Liao et al. proposed an adaptive LSB substitution based on PVD [12]. As per their proposal, in a 2×2 pixel block, the average pixel value difference is to be calculated; if this value is smaller, then less number of LSBs can be used for data hiding; otherwise more number of LSBs can be used for data hiding. This approach was further extended to 3×3 size blocks to achieve higher embedding capacity by Swain [13]. To improve the hiding capacity and un-detectability, Wu and Tsai's PVD technique has been extended to seven-directional PVD [14] using 3×3 size pixel blocks. Another type of multi-directional PVD with LSB substitution was also proposed by Darabkh et al. [15]. Khodaei and Faez's PVD and LSB approach is extended to 3×3 size blocks to exploit multi-directional edges so that higher hiding capacity and protection against RS analysis and PDH analysis are achieved [16]. Jung [17] used LSB substitution

✉ Gandharba Swain
gswain1234@gmail.com

[1] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh 522502, India

Table 1 Range table

Range	$r_1 = [0, 7]$	$r_2 = [8, 15]$	$r_3 = [16, 31]$	$r_4 = [32, 63]$
Capacity, n_j	3	3	4	5

in lower bit planes and PVD in higher bit planes of a pixel in a block. Yang et al. used 4 pixel blocks and grouped two pixels in varieties of ways like horizontal, vertical, and diagonal directions to achieve improved performance [18]. Nilizadeh and Nilchi proposed a PVD technique based on smoothness and complexity in neighborhood of a pixel, wherein the user can select the size of the square blocks [19].

There exist a wide range of applications of image steganography techniques. Yesilyurt and Yalman used steganography to protect the stored data elements on cloud [20]. Xiang et al. proposed the usage of steganography for sending data from a mobile to cloud and vice versa [21]. Reversible data hiding has been used by Singh and Raman to protect the unauthorized usage of data stored in cloud [22]. Wu et al. also proposed use of steganography for authentication and secret sharing [23]. Steganography can also be used for secured localization of nodes in wireless sensor networks [24].

The existing PVD steganography techniques possess a maximum hiding capacity approximately 4.0 bits per a byte (BPB). This article proposes a data hiding technique with very high hiding capacity i.e. 4.55 BPB. So that more than 55% size of the image can be used to hide the data.

2 Related Work

Jung [17] has proposed a data hiding technique by mixing LSB substitution and PVD in a pixel. The image is divided into blocks, each block with two consecutive pixels, (P_1, P_2). LSB substitution is applied on k LSBs of P_1 and P_2. PVD substitution is applied on remaining (8-k) bits of P_1 and P_2. The embedding procedure is as narrated below.

The quotient block, i.e., higher bit-plane block, (Q_1, Q_2) is derived from the pixel block by using quotient division on P_1 and P_2 as shown in Eq. (1). Similarly the LSB block, i.e., lower bit-plane block, (R_1, R_2) is derived by using remainder division on P_1 and P_2 as in Eq. (2).

$$Q_1 = P_1 \text{ div } 2^k \text{ and } Q_2 = P_2 \text{ div } 2^k \tag{1}$$

$$R_1 = P_1 \bmod 2^k \text{ and } R_2 = P_2 \bmod 2^k \tag{2}$$

Here div and mod are the quotient and remainder division operators, respectively. For example: $5 \text{ div } 2 = 2$ *and* $5 \bmod 2 = 1$.

The difference value, $d = |Q_1 - Q_2|$, is calculated. It belongs to one of the ranges of the range Table 1. Suppose the lower bound of that range is L_j and hiding capacity is n_j (for some value of j, $1 \leq j \leq 4$). From secret binary data stream, n_j bits of data are taken and converted to decimal value, S.

The new difference values, $d' = L_j + S$ and $m = |d' - d|$, are calculated. Suppose the stego-values of Q_1 and Q_2 are Q_1' and Q_2', respectively. Q_1' and Q_2' are calculated as in Eq. (3).

$$(Q_1', Q_2') = \begin{cases} (Q_1 - \lceil m/2 \rceil, Q_2 + \lfloor m/2 \rfloor), & \text{if } d \text{ is odd} \\ (Q_1 - \lfloor m/2 \rfloor, Q_2 + \lceil m/2 \rceil), & \text{if } d \text{ is even} \end{cases} \tag{3}$$

Now, k bits of binary data are taken from the secret binary data stream and converted to decimal value, R_1'. Again another k bits of binary data are taken from the secret data stream and converted to decimal value, R_2'. R_1' and R_2' are the stego-values of R_1 and R_2, respectively. The stego-pixel pair (P_1, P_2) is formed using Eq. (4).

$$(P_1', P_2') = \left(Q_1' \times 2^k + R_1', Q_2' \times 2^k + R_2'\right) \tag{4}$$

The extraction from the block (P_1', P_2') is done in the following way. Quotient and remainder are calculated using Eqs. (5) and (6), respectively.

$$Q_1' = \left(P_1' - \left(P_1' \bmod 2^k\right)\right) \text{ div } 2^k \text{ and}$$

$$Q_2' = \left(P_2' - \left(P_2' \bmod 2^k\right)\right) \text{div} 2^k \tag{5}$$

$$R_1' = P_1' \bmod 2^k \text{ and } R_2' = P_2' \bmod 2^k \tag{6}$$

The difference value, $d' = |Q_1' - Q_2'|$, is again calculated. The difference d' belongs to one of the ranges of range Table 1, whose hiding capacity is n_j and lower bound is L_j. The decimal value of the embedded secret bits in the two quotients is b. It is calculated as $b = d' - L_j$. This b is converted to n_j bits in binary and appended to extracted binary data stream. R_1' is converted to k bits in binary and appended to the extracted binary data stream. Similarly, R_2' is converted to k bits in binary and appended to the extracted binary data stream.

The fall off boundary problem (FOBP) in Jung's technique has been illustrated in the following sentences. Let us consider, $k = 2$. Consider a block with two pixels, $P_1 = 129$ and $P_2 = 249$. Using Eqs. (1) and (2), $(Q_1, Q_2) = (32, 62)$ and $(R_1, R_2) = (1, 1)$ are calculated. The difference value $d = |32 - 62| = 30 \epsilon r_3$, so $n_j = 4$ and $L_j = 16$. Suppose that the secret binary data to be embedded is:

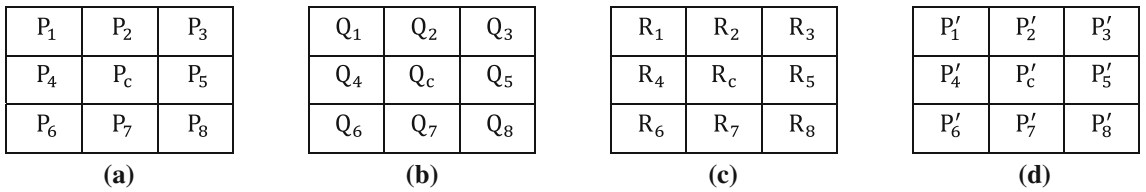

Fig. 1 **a** Original block, **b** quotient block, **c** remainder block, and **d** stego-block

$(00100001)_2$. From this secret binary data stream 4-bits of data are taken and converted to decimal value, i.e., $(0010)_2$ is converted to decimal value 2. Hence S=2. The new difference value, $d' = L_j + S = 16 + 2 = 18$, is obtained. $m = |d' - d| = |18 - 30| = 12$ is obtained. The value of d is 30, an even number, hence using Eq. (3), $Q'_1 = (Q_1 - \lfloor m/2 \rfloor) = 26$ and $Q_2 = (Q_2 + \lceil m/2 \rceil) = 68$. From secret binary data stream 2-bits of data are taken, it is $(00)_2$, its decimal value is 0, so $R'_1 = 0$. Again, 2-bits of data are taken, it is $(01)_2$, its decimal value is 1, so $R'_2 = 1$. The stego-pixels P'_1 and P'_2 are computed using Eq. (4) as $P'_1 = 26 \times 4 + 0 = 104$ and $P'_2 = 68 \times 4 + 1 = 273$. *The stego-pixel P'_2 is exceeding the upper boundary value 255. Thus Jung's technique suffers from FOBP.* The proposed technique takes care of the FOBP.

3 Proposed Eight Directional QVD Technique

The image is partitioned into non-overlapping blocks consisting of 3×3 pixels. A sample block is as shown in Fig. 1a. The embedding procedure is described by the following steps.

Step 1 From this pixel block, the quotient block (i.e., higher bit-plane block) is formed by applying quotient division on every pixel using Eq. (7); this block is shown in Fig. 1b. Similarly the LSB block (i.e., lower bit-plane block) is formed by applying remainder division on every pixel using Eq. (8); this block is shown in Fig. 1c.

$$Q_i = P_i \text{ div } 4, \text{ for } i = 1 \text{ to } 8 \quad \text{and} \quad Q_c = P_c \text{ div } 4 \tag{7}$$

$$R_i = P_i \bmod 4, \text{ for } i = 1 \text{ to } 8 \quad \text{and} \quad R_c = P_c \bmod 4 \tag{8}$$

Here the operators div and *mod* stand for quotient and remainder division, respectively. For example: 5 div 2 = 2 and 5 mod 2 = 1.

Step 2 Suppose the two LSBs of P_c are t_2t_1. R_c is the decimal equivalent of t_2t_1. R_c is converted to R'_c after substituting 1_2 in place of t_1 and a bit from binary data stream in t_2. Similarly, for $i = 1$ to 8, two binary bits are taken from binary data stream, converted to decimal, and it is named as R'_i, the stego-value of R_i.

Step 3 From the quotients there are eight pairs. Those are (Q_c, Q_1), (Q_c, Q_2), (Q_c, Q_3), (Q_c, Q_4), (Q_c, Q_5), (Q_c, Q_6), (Q_c, Q_7), and (Q_c, Q_8). Eight difference values are calculated using Eq. (9).

$$d_i = Q_i - Q_c, \text{ for } i = 1 \text{ to } 8 \tag{9}$$

Table 1 is used as range table. Suppose the absolute difference values $|d_i|$, for $i = 1$ to 8 falls into range r_j (for $1 \le j \le 4$) in the range table, whose lower bound is L_{ji} and hiding capacity is n_{ji}.

Step 4 For $i = 1$ to 8; n_{ji} data bits are collected and converted to decimal values b_i. Then the new difference values d'_i, are calculated using Eq. (10).

$$d'_i = \begin{cases} L_{ji} + b_i, & \text{if } \ d_i \ge 0, \\ -L_{ji} - b_i, & \text{if } \ d_i < 0 \end{cases} \tag{10}$$

Now the values of m_i, for $i = 1$ to 8 are computed as in (11).

$$m_i = d'_i - d_i \tag{11}$$

Step 5 The eight quotient pairs are denoted as (Q_c, Q_i), for $i = 1$ to 8. Now embedding into the eight pairs is done as in Eq. (12).

$$(Q'_{ci}, Q'_i) = \begin{cases} Q_c - \lfloor m_i/2 \rfloor, Q_i + \lceil m_i/2 \rceil), & \text{if } d_i \text{ is even}, \\ Q_c - \lceil m_i/2 \rceil, Q_i +' \lfloor m_i/2 \rfloor), & \text{if } d_i \text{ is odd} \end{cases} \tag{12}$$

After embedding, the pixel Q_c has got eight new values in eight different pairs. So they are to be unified into a single value using Eq. (13).

$$Q'_c = \left\lceil \frac{(Q'_{c1} + Q'_{c2} + Q'_{c3} + Q'_{c4} + Q'_{c5} + Q'_{c6} + Q'_{c7} + Q'_{c8})}{8} \right\rceil \tag{13}$$

As each Q'_{ci} has to be changed to Q'_c, at the same time the corresponding Q'_i is changed by using Eq. (14).

$$Q'_i = Q'_i + (Q'_c - Q'_{ci}), \text{ for } i = 1 \text{ to } 8 \tag{14}$$

The stego-pixel values are calculated using Eq. (15), and the stego-block is as shown in Fig. 1d.

$$P'_c = Q'_c \times 4 + R'_c, \; P'_i = Q'_i \times 4 + R'_i, \text{ for } i = 1 \text{ to } 8 \tag{15}$$

130	132	130
128	128	130
131	129	128

(a)

32	33	32
32	32	32
32	32	32

(b)

2	0	2
0	0	2
3	1	0

(c)

125	147	138
124	127	133
137	144	151

(d)

Fig. 2 **a** Original block, **b** quotient block, **c** remainder block, and **d** stego-block

Step 6 If any one of the Q_i' value falls off the boundary $\{0, 63\}$, then in the whole block QVD embedding is undone and the 4-bit LSB substitution is applied as below.

In first LSB bit of P_c, bit 0 is embedded. In other three LSBs, three data bits are embedded. In all the remaining pixels P_i, for $i = 1$ to 8, 4-bit LSB substitution is applied. After applying 4-bit LSB substitution suppose the pixel values are P_c', P_1', P_2', P_3', P_4', P_5', P_6', P_7' and P_8'. Suppose the decimal value of 4 LSBs in P_c is dec_{old} and the decimal value of 4 LSBs in P_c' is dec_{new}. Now find the deviation, $\text{dev} = \text{dec}_{\text{old}} - \text{dec}_{\text{new}}$. The modified value of P_c' is calculated using Eq. (16). Similarly, suppose for $i = 1$ to 8, the decimal value of 4 LSBs in P_i is deci_{old} and the decimal value of 4 LSBs in P_i' is deci_{new}. Now find the deviation, $\text{devi} = \text{deci}_{\text{old}} - \text{deci}_{\text{new}}$. The modified value of P_i' is calculated using Eq. (17).

$$P_c' = \begin{cases} P_c' + 2^5, & \text{if} \quad \text{dev} > 2^4 \text{ and } 0 \le P_c' + 2^5 \le 255 \\ P_c' - 2^5, & \text{if} \quad \text{dev} < -2^4 \text{ and } 0 \le P_c' - 2^5 \le 255 \\ P_c', & \text{otherwise} \end{cases} \tag{16}$$

$$P_i' = \begin{cases} P_i' + 2^5, & \text{if} \quad \text{devi} > 2^4 \text{ and } 0 \le P_i' + 2^5 \le 255 \\ P_i' - 2^5, & \text{if} \quad \text{devi} < -2^4 \text{ and } 0 \le P_i' - 2^5 \le 255 \\ P_i', & \text{otherwise} \end{cases} \tag{17}$$

The stego-pixel values are as shown in Fig. 1d.

The extraction procedure is described by the following steps.

Step 1 Suppose the stego-pixel block is as shown in Fig. 1d. From the central pixel, P_c' check the LSB bit, if it is 0, then in this block 4-bit LSB substitution was applied during embedding. If it is 1, then 2-bit LSB substitution and QVD approach was applied during embedding.

Step 2 If it is 4-bit LSB substitution, then extract three next LSBs of P_c' and 4-LSBs from each P_i', for $i = 1$ to 8. Append all these 35 bits to the extracted bit stream.

Step 3 If it is not 4-bit LSB, it is 2-bit LSB and QVD. Then extract the second LSB from P_c' and 2-LSBs from each P_i', for $i = 1$ to 8. Append these 17 bits to extracted binary data stream. The two LSBs from each P_i' can be extracted by calculating $R_i' = P_i' \bmod 4$ and converting each R_i' to two binary bits. Furthermore, calculate the quotients as in Eq. (18).

$$Q_i' = P_i' \text{ div } 4, \text{ for } i = 1 \text{ to } 8 \text{ and } Q_c' = P_c' \bmod 4 \tag{18}$$

Now calculate eight difference values using Eq. (19)

$$d_i' = |Q_c' - Q_i'|, \text{ for } i = 1 \text{ to } 8 \tag{19}$$

Each d_i' belongs to a range in range Table 1, whose embedding capacity is n_{ji} and lower bound is L_{ji}.

Now for each d_i', the extracted value in decimal is b_i' calculated as in Eq. (20).

$$b_i' = |d_i' - L_{ji}|, \text{ for } i = 1 \text{ to } 8 \tag{20}$$

For $i = 1$ to 8, each b_i' is converted to n_{ji} binary bits and these binary bits are appended to extracted binary data stream. Thus, extraction is completed.

4 Example of Embedding and Extraction

To understand the embedding procedure, let us use the 3×3 block given in Fig. 2a. The central pixel, $P_c = 128$ and its neighboring pixels are $P_1 = 130$, $P_2 = 132$, $P_3 = 130$, $P_4 = 128$, $P_5 = 130$, $P_6 = 131$, $P_7 = 129$, $P_8 = 128$.

Step 1 After applying Eq. (7) the quotients are $Q_1 = 32$, $Q_2 = 33$, $Q_3 = 32$, $Q_4 = 32$, $Q_5 = 32$, $Q_6 = 32$, $Q_7 = 32$, $Q_8 = 32$, and $Q_c = 32$. Similarly by applying Eq. (8), the remainders are $R_1 = 2$, $R_2 = 0$, $R_3 = 2$, $R_4 = 0$, $R_5 = 2$, $R_6 = 3$, $R_7 = 1$, $R_8 = 0$, and $R_c = 0$.

Step 2 The binary value of P_c is 10000000. The two LSBs of P_c are 00. Suppose the secret binary data stream to be embedded into the block is: 1 01 11 10 00 01 01 00 11 000 101 011 000 010 011 101 110. The first LSB of P_c is replaced by 1, to act as indicator at receiver and the second LSB should be replaced by a data bit taken from the secret data stream. Thus, the two new LSBs of P_c are 11. So, $R_c' = 3$. Take next two bits of data from binary data stream and convert to decimal value, so $R_1' = 1$. Again, take next two bits of data from binary data stream and convert to decimal value, so $R_2' = 3$. Similarly, $R_3' = 2$, $R_4' = 0$, $R_5' = 1$, $R_6' = 1$, $R_7' = 0$, and $R_8' = 3$.

Step 3 The quotients form eight pairs (Q_c, Q_i), for $i = 1$ to 8. Those are (32, 32), (32, 33), (32, 32), (32, 32), (32, 32), (32, 32), (32, 32), and (32, 32). Using Eq. (9), eight difference values are $d_1 = 0$, $d_2 = 1$, $d_3 = 0$, $d_4 = 0$, $d_5 = 0$, $d_6 = 0$, $d_7 = 0$, and $d_8 = 0$. All these difference values belong to range R_1 in Table 1. So the hiding capacities in the eight

directions are $n_{j1} = 3$, $n_{j2} = 3$, $n_{j3} = 3$, $n_{j4} = 3$, $n_{j5} = 3$, $n_{j6} = 3$, $n_{j7} = 3$, and $n_{j8} = 3$. The respective lower bounds are $L_{j1} = 0$, $L_{j2} = 0$, $L_{j3} = 0$, $L_{j4} = 0$, $L_{j5} = 0$, $L_{j6} = 0$, $L_{j7} = 0$, and $L_{j8} = 0$.

Step 4 Take next 3 bits of data from binary data stream and convert to decimal value, so $b_1 = 0$. Take next 3 bits of data from binary data stream and convert to decimal value, so $b_2 = 5$. Similarly, $b_3 = 3$, $b_4 = 0$, $b_5 = 2$, $b_6 = 3$, $b_7 = 5$, and $b_8 = 6$. Using Eq. (10), new difference values are $d'_1 = 0, d'_2 = 5, d'_3 = 3, d'_4 = 0, d'_5 = 2, d'_6 = 3, d'_7 = 5$, and $d'_8 = 6$. Now using Eq. (11), $m_1 = 0, m_2 = 4, m_3 = 3$, $m_4 = 0, m_5 = 2, m_6 = 3, m_7 = 5$, and $m_8 = 6$.

Step 5 Applying Eq. (12), the eight stego-quotient pairs are:$(Q'_{c1}, Q'_1) = (32, 32)$, $(Q'_{c2}, Q'_2) = (30, 35)$, $(Q'_{c3}, Q'_3) = (31, 34)$, $(Q'_{c4}, Q'_4) = (32, 32)$, $(Q'_{c5}, Q'_5) = (31, 33)$, $(Q'_{c6}, Q'_6) = (31, 34)$, $(Q'_{c7}, Q'_7) = (30, 35)$, and $(Q'_{c8}, Q'_8) = (29, 35)$. Using Eq. (13), $Q'_c = (32 + 30 + 31 + 32 + 31 + 31 + 30 + 29)/8 = 31$. As each Q'_{ci} has to be changed to Q'_c, at the same time Q'_i is changed by using Eq. (14). Thus,$Q'_1 = 31$, $Q'_2 = 36$, $Q'_3 = 34$, $Q'_4 = 31$, $Q'_5 = 33$, $Q'_6 = 34$, $Q'_7 = 36$, and $Q'_8 = 37$. Using Eq. (15), stego-pixel values are $P'_c = 127$, $P'_1 = 125$, $P'_2 = 147$, $P'_3 = 138$, $P'_4 = 124$, $P'_5 = 133$, $P'_6 = 137$, $P'_7 = 144$, and $P'_8 = 151$. The stego-pixel block is shown in Fig. 2d.

Step 6 All the calculated Q'_i values are in range $\{0, 63\}$, so FOBP did not occur.

To understand the extraction procedure, let us apply it on stego-block shown in Fig. 2d.

Step 1 $P'_c = 127$, $P'_1 = 125$, $P'_2 = 147$, $P'_3 = 138$, $P'_4 = 124$, $P'_5 = 133$, $P'_6 = 137$, $P'_7 = 144$, and $P'_8 = 151$. In binary, $P'_c = 01111111$. The LSB bit is 1, so 2-bit LSB substitution and QVD was applied during embedding.

Step 2 Not applicable

Step 3 Extract the second LSB from P'_c and append to extracted data stream (EDS). Thus, EDS = 1. Calculate the remainders $R'_i = P'_i \bmod 4$. Thus, $R'_1 = 1$, $R'_2 = 3$, $R'_3 = 2$, $R'_4 = 0$, $R'_5 = 1$, $R'_6 = 1$, $R'_7 = 0$, and $R'_8 = 3$. Converting each of these values to 2 binary bits and appending to EBS, we get EDS = 1 01 11 10 00 01 01 00 11.

Using Eq. (18), the quotients are $Q'_c = 31, Q'_1 = 31, Q'_2 = 36, Q'_3 = 34, Q'_4 = 31, Q'_5 = 33, Q'_6 = 34$, $Q'_7 = 36$, and $Q'_8 = 37$. Using Eq. (19), $d'_1 = 0$, $d'_2 = 5$, $d'_3 = 3$, $d'_4 = 0$, $d'_5 = 2$, $d'_6 = 3$, $d'_7 = 5$, and $d'_8 = 6$. Each d'_i belongs to a range R_1, so $L_{j1} = 0$, $L_{j2} = 0, L_{j3} = 0, L_{j4} = 0$, $L_{j5} = 0$, $L_{j6} = 0$, $L_{j7} = 0$, and $L_{j8} = 0$. Also, $n_{j1} = 3$, $n_{j2} = 3, n_{j3} = 3$, $n_{j4} = 3$, $n_{j5} = 3$, $n_{j6} = 3$, $n_{j7} = 3$, and $n_{j8} = 3$.

Using Eq. (20), $b'_1 = 0$, $b'_2 = 5$, $b'_3 = 3$, $b'_4 = 0$, $b'_5 = 2$, $b'_6 = 3$, $b'_7 = 5$, and $b'_8 = 6$.

Each b'_i is converted to n_i binary bits and appended to EDS. Thus, EDS = 1 01 11 10 00 01 01 00 11 000 101 011 000 010 011 101 110. This is the extracted data in total. This data are same as the embedded data stream.

5 Results and Discussion

The proposed technique is implemented using MATLAB R2017b. The original RGB color images used for testing are gathered from SIPI image database [25]. Figure 3 represents a set of input images. Seven lakhs (7,00,000) bits of secret binary data are hidden in these test images and their respective stego-images are shown in Fig. 4. The resultant stego-images are innocent and do not attract any attention of an intruder.

The effectiveness of the proposed data hiding technique is evaluated by the metrics like, hiding capacity (HC), bits per byte (BPB), peak signal-to-noise ratio (PSNR), and quality index (QI). HC refers to the maximum number of data bits that can be concealed inside an image. Sometimes, we represent the hiding capacity as BPB, i.e., the average hiding capacity per a byte of the image. PSNR is a metric to evaluate the distortion in a stego-image. It can be calculated as in Eq. (21), where X_{ij} is original image pixel and Y_{ij} is stego-image pixel at coordinate (i, j).

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{m \times n \times 255 \times 255}{\sum_{i=1}^{m} \sum_{j=1}^{n} (X_{ij} - Y_{ij})} \tag{21}$$

QI is a metric to estimate the similarity between original and stego-image. It is measured as in Eq. (22).

$$\mathrm{QI} = \frac{4 \times \bar{X} \times \bar{Y} \times \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} (X_{ij} - \bar{X}) \times (Y_{ij} - \bar{Y}) \right\}}{\left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} (X_{ij} - \bar{X})^2 + \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - \bar{Y})^2 \right\} \times \left\{ (\bar{X})^2 + (\bar{Y})^2 \right\}} \tag{22}$$

Where $\bar{X}$ is the average pixel value of original image, and $\bar{Y}$ is the average pixel value of stego-image.

The performance of Khodaei and Faez's technique [10] and Pradhan et al.'s technique [14] is shown in Table 2. Furthermore, the performance of Swain's technique [16] and the proposed technique is shown in Table 3. From these two tables, it can be analyzed that the HC of the proposed technique is higher than that of the three existing techniques. The BPB of the proposed technique is 4.55; it means that in every 8-bits of the image 4.55 bits of data can be hidden. This is a very great achievement. A BPB of 4.55 was never found in the literature.

The performance of the proposed technique has been compared with the techniques in [27–30]. The steganography technique proposed in [27] uses hybrid adaptive neural networks and modified genetic algorithm. To protect from security analysis multiple security layers have been incorporated. The hiding capacity is 3.8 BPB with PSNR greater than 40 dB. In [28] a steganography technique with three phases using adaptive neural networks and adaptive genetic algorithm has been proposed. The hiding capacity is up to 4

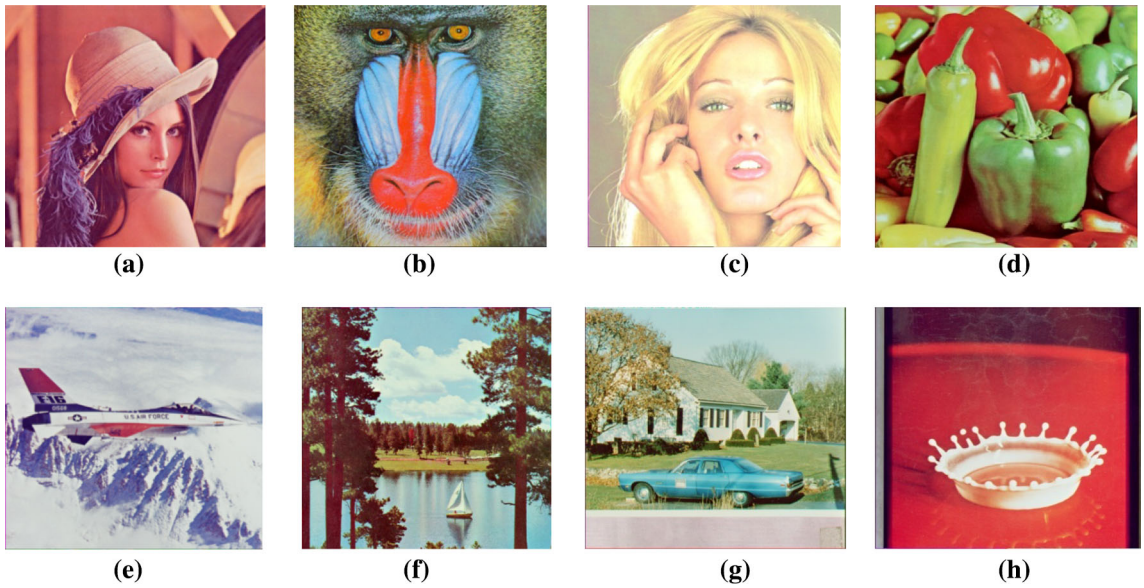

Fig. 3 Original images. **a** Lena, **b** Baboon, **c** Tiffany, **d** Peppers, **e** Jet, **f** Boat, **g** House, **h** Pot

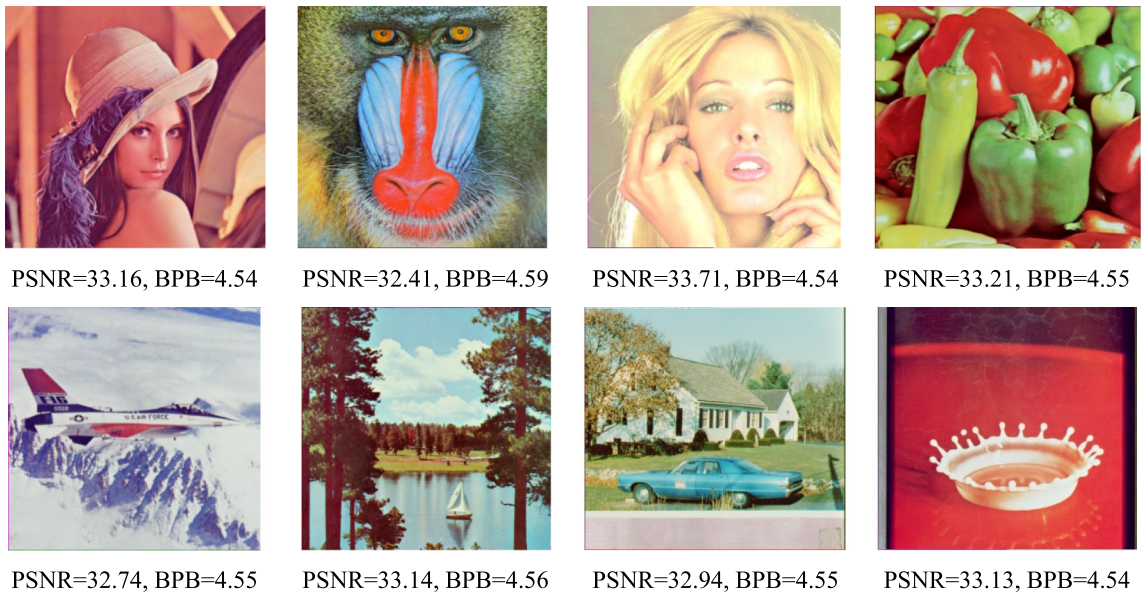

Fig. 4 Stego-images

BPB. With hiding capacity of 4 BPB, the structural similarity index (SSIM) is 0.9997. The steganography technique in [29] uses image segmentation and adaptive neural networks. The recorded hiding capacity is 2 BPB. With 25% of hiding capacity the recorded PSNR is greater than 40 dB. In [30] an image steganography technique using image segmentation and modified least significant bit (M-LSB) substitution has been proposed. The hiding capacity is very good, and it was recorded to be 4 BPB with PSNR greater than 40 dB. As it uses M-LSB substitution, it is not secured and can be detected by RS analysis. The proposed technique possesses 4.55 BPB, with a reduced PSNR of 33.06. Although a PSNR value greater than 40 dB is appreciable, but also a PSNR value in between 30 and 40 dB is acceptable.

6 Security Analysis

The proposed data hiding technique has been analyzed by PDH analysis and RS analysis. The PDH analysis has been done in the following way. In the image, pixels are grouped into different groups of two consecutive pixels. In every group, their difference value is calculated [26]. Each of these difference values falls in range -255 to $+255$. If there are n

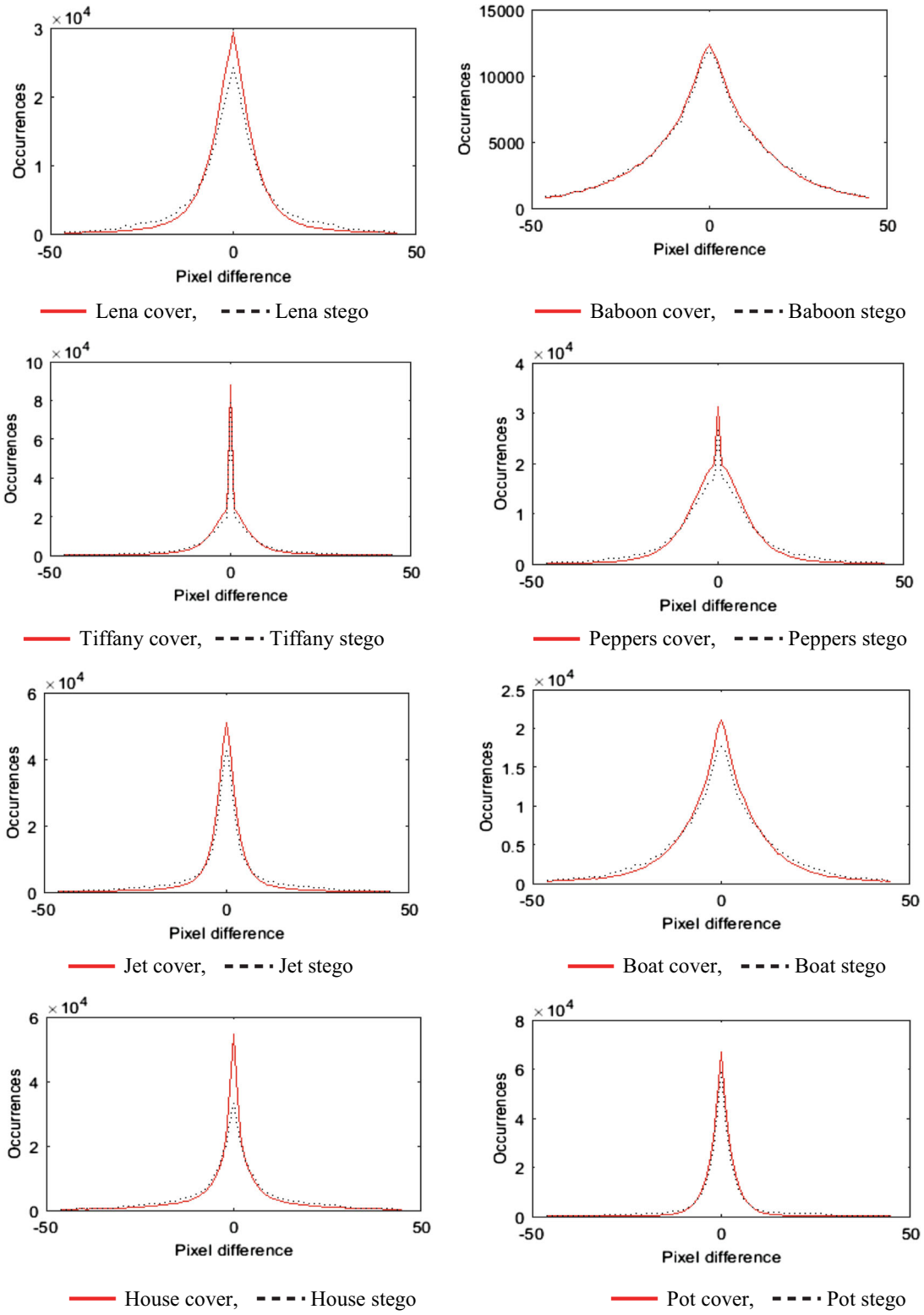

Fig. 5 PDH analysis of the proposed technique

Table 2 Results of existing techniques

Images 512×512×3	Khodaei and Faez-type 2 [10]				Pradhan et al.'s PVD [14]			
	PSNR	HC	QI	BPB	PSNR	HC	QI	BPB
Lena	41.25	2,434,603	0.9993	3.09	41.73	1,901,149	0.9993	2.41
Baboon	34.49	2,662,080	0.9963	3.38	33.77	2,243,218	0.9957	2.85
Tiffany	40.25	2,416,944	0.9982	3.07	41.30	1,450,799	0.9986	1.84
Peppers	37.91	2,435,223	0.9987	3.09	40.20	1,806,166	0.9992	2.29
Jet	40.64	2,418,419	0.9985	3.07	41.98	1,909,595	0.9989	2.42
Boat	37.49	2,504,613	0.9987	3.18	37.91	1,991,005	0.9988	2.53
House	38.75	2,470,824	0.9985	3.14	38.98	1,977,403	0.9986	2.51
Pot	37.83	2,387,494	0.9990	3.03	42.40	1,803,635	0.9996	2.29
Average	38.57	2,466,275	0.9984	3.13	39.78	1,885,371	0.9985	2.39

Table 3 Results of Swain's PVD technique and proposed QVD + LSB technique

Images 512×512×3	Swain's PVD-type 2 [16]				Proposed QVD + LSB technique			
	PSNR	HC	QI	BPB	PSNR	HC	QI	BPB
Lena	40.48	2,533,551	0.9992	3.22	33.16	3,573,202	0.9956	4.54
Baboon	32.09	2,939,376	0.9937	3.74	32.41	3,615,390	0.9941	4.59
Tiffany	40.53	2,511,139	0.9984	3.19	33.71	3,572,814	0.992	4.54
Peppers	34.69	2,544,392	0.9975	3.24	33.21	3,576,879	0.9965	4.55
Jet	40.38	2,538,801	0.9985	3.23	32.74	3,576,372	0.9912	4.55
Boat	34.08	2,659,795	0.9973	3.38	33.14	3,585,039	0.9966	4.56
House	38.39	2,625,804	0.9984	3.34	32.94	3,578,488	0.9947	4.55
Pot	37.11	2,475,977	0.9988	3.15	33.13	3,572,977	0.9971	4.54
Average	37.22	2,603,604	0.9977	3.31	33.06	3,581,395	0.9947	4.55

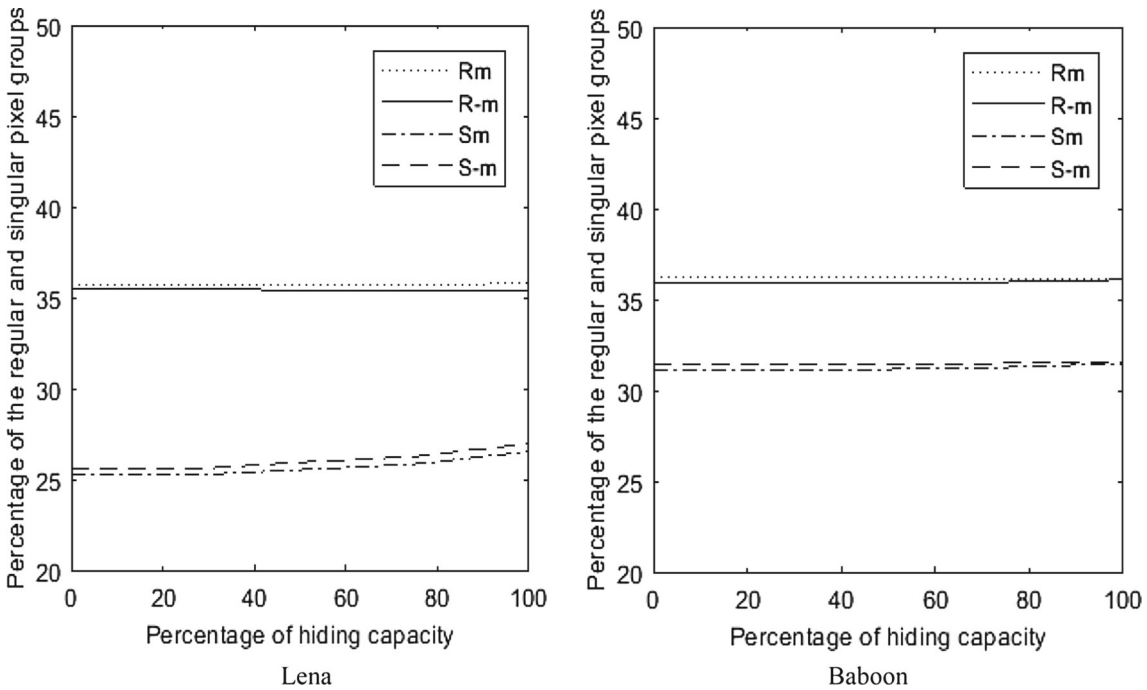

Fig. 6 RS analysis of the proposed technique

pixels in an image, then $n/2$ difference values are obtained. A graph is plotted with the frequency of various difference values on Y-axis and difference on X-axis. Such a graph can be called as a PDH graph. It is known that the PDH of any original image will be a smooth curve. If the stego-image is more distorted, its PDH curve becomes zig-zag in nature. If the distortion in the stego-image is very less, then its PDH curve looks smooth like that of an original image. The zig-zag nature of PDH curve is known as the step effect.

The PDH analysis for the proposed technique with 8 test images is shown in eight sub-figures of Fig. 5. In each sub-figure the solid line curve stands for the original image and the dotted line curve stands for the stego-image. From these eight sub-figures, it could be analyzed that the PDH curves of the stego-images do not show any step effects. Therefore, it is concluded that the proposed technique is undetectable by PDH analysis.

Since the proposed technique uses LSB substitution in lower bit planes, so it should be passed through RS analysis. To perform RS analysis two functions, (i) F_1: $2n \leftrightarrow 2n+1$, and (ii) F_{-1}: $2n \leftrightarrow 2n-1$ are used. The first function represents changes from value $2n$ to value $2n+1$ and vice-versa. Similarly, the second function represents changes from value 2n to value $2n-1$, and from value $2n-1$ to value $2n$. The image M is partitioned into a number of equal size blocks. Say, B is a block and the pixels in B are $x_1, x_2, x_3, \ldots, x_n$. To measure the smoothness of B, the function F is defined as $F(x_1, x_2, x_3, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$. Then apply F_1 and F_{-1} to all the blocks of M and define the four parameters R_m, S_m, R_{-m}, and S_{-m}. R_m is a fraction of the total number of blocks satisfying the condition $F(F_1(B)) > F(B)$, and S_m is a fraction of the total number of blocks satisfying the condition $F(F_1(B)) < F(B)$. Similarly, R_{-m} is a fraction of the total number of blocks satisfying the condition $F(F_{-1}(B)) > F(B)$, and S_{-m} is a fraction of the total number of blocks satisfying the condition $F(F_{-1}(B)) < F(B)$. If $R_m \approx R_{-m} > S_m \approx S_{-m}$ is true, then RS analysis fails to detect the steganography technique. But if the condition $R_{-m} - S_{-m} > R_m - S_m$ is true, then the RS analysis succeeds in detecting the steganography technique.

The two sub-figures in Fig. 6 represent the RS analysis of the proposed technique over Lena and Baboon images. Lena is a smoother image, and Baboon is a textured image. From Fig. 6 it could be observed that for the proposed technique, $R_m \approx R_{-m} > S_m \approx S_{-m}$ satisfies for the edged image, Baboon and smoother image, Lena. As Lena is the smoother image and Baboon is the textured image, it can be predicted that the results of all the remaining images will fall in between these two. So it is worthy to mention that the proposed technique is undetectable by RS analysis.

7 Conclusion

This research article proposes a data hiding technique in images with higher embedding capacity using a combination of LSB substitution and quotient value differencing (QVD). It operates on the image by splitting it into non-overlapped 3×3 pixel blocks. In every block LSB substitution is applied at lower bit planes and QVD is applied at higher bit planes. Every block after embedding is checked for FOBP. If it suffers with FOBP, then the proposed embedding is undone and direct 4-bit LSB substitution is applied. However, such FOBP cases are very less in number. Experimentally, it is evidenced that the hiding capacity is improved to a greater extent. As this technique uses LSB substitution, and quotient value differencing, its efficacy is tested by both PDH analysis and RS analysis. It is found that these two steganalysis techniques cannot detect the proposed steganography technique.

References

1. Cheddad, A.; Condell, J.; Curran, K.; Kevitt, P.M.: Digital image steganography: survey and analysis of current methods. Signal Process. **90**, 727–752 (2010)
2. Martin, A.; Sapiro, G.; Seroussi, G.: Is image steganography natural? IEEE Trans. Image Process. **14**(12), 2040–2050 (2005)
3. Fridrich, J.; Goljian, M.; Du, R.: Detecting LSB Steganography in color and gray-scale images. Mag. IEEE Multimed. Secur. **8**(4), 22–28 (2001)
4. Wu, D.C.; Tsai, W.H.: A steganograhic method for images by pixel value differencing. Pattern Recogn. Lett. **24**(9), 1613–1626 (2003)
5. Zhang, X.; Wang, S.: Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security. Pattern Recogn. Lett. **25**, 331–339 (2004)
6. Chang, K.C.; Chang, C.P.; Huang, P.S.; Tu, T.M.: A novel image steganography method using tri-way pixel value differencing. J. Multimed. **3**(2), 37–44 (2008)
7. Lee, Y.P.; Lee, J.C.; Chen, W.K.; Chang, K.C.; Su, I.J.; Chang, C.P.: High-payload image hiding with quality recovery using tri-way pixel-value differencing. Inf. Sci. **191**, 214–225 (2012)
8. Wu, H.C.; Wu, N.I.; Tsai, C.S.; Hwang, M.S.: Image steganographic scheme based on pixel-value differencing and LSB replacement methods. IEEE Proc. Vis. Image Signal Process. **152**(5), 611–615 (2005)
9. Yang, C.H.; Weng, C.Y.; Wang, S.J.; Sun, H.M.: Varied PVD + LSB evading programs to spatial domain in data embedding systems. J. Syst. Softw. **83**(10), 1635–1643 (2010)
10. Khodaei, M.; Faez, K.: New adaptive steganographic method using least-significant-bit substitution and pixel-value differencing. IET Image Proc. **6**(6), 677–686 (2012)
11. Swain, G.: A steganographic method combining LSB substitution and PVD in a block. Proc. Comput. Sci. **85**, 39–44 (2016)
12. Liao, X.; Wen, Q.Y.; Zhang, J.: A steganographic method for digital images with four-pixel differencing and modified LSB substitution. J. Vis. Commun. Image Represent. **22**(1), 1–8 (2011)
13. Swain, G.: Digital image steganography using nine-pixel differencing and modified LSB substitution. Indian J. Sci. Technol. **7**(9), 1444–1450 (2014)
14. Pradhan, A.; Sekhar, K.R.; Swain, G.: Digital image steganography based on seven way pixel value differencing. Indian J. Sci. Technol. **9**(37), 1–11 (2016)

15. Darabkh, K.A.; Al-Dhamari, A.K.; Jafar, I.F.: A new steganographic algorithm based on multi directional PVD and modified LSB. J. Inf. Technol. Control **46**(1), 16–36 (2017)
16. Swain, G.: Digital image steganography using eight directional PVD against RS analysis and PDH analysis. Adv. Multimed. (2018). **(in press)**
17. Jung, K.H.: Data hiding scheme improving embedding capacity using mixed PVD and LSB on bit plane. J. Real Time Image Process. **14**(1), 127–136 (2018)
18. Yang, C.H.; Weng, C.Y.; Tso, H.K.; Wang, S.J.: A data hiding scheme using the varieties of pixel-value differencing in multimedia images. J. Syst. Softw. **84**, 669–678 (2011)
19. Nilizadeh, A.F.; Nilchi, A.R.: Block texture pattern detection based on smoothness and complexity of neighborhood pixels. Int. J. Image Graph. Signal Process. **5**, 1–9 (2014)
20. Yesilyurt, M.; Yalman, Y.: New approach for cloud computing security: using data hiding methods. Sadhana **41**(11), 1289–1298 (2016)
21. Xiang, T.; Hu, J.; Sun, J.: Outsourcing chaotic selective image encryption to the cloud with steganography. Digit. Signal Proc. **43**, 28–37 (2015)
22. Singh, P.; Raman, B.: Reversible data hiding based on Shamir's secret sharing for color images over cloud. Inf. Sci. **422**, 77–97 (2018)
23. Wu, C.C.; Kao, S.J.; Hwang, M.S.: A high quality image sharing with steganography and adaptive authentication scheme. J. Syst. Softw. **84**, 2196–2207 (2011)
24. Tondwalkar, A.; Jani, P.V.: Secure localization of wireless devices with application to sensor networks using steganography. Proc. Comput. Sci. **78**, 610–616 (2016)
25. USC-SIPI Image Database (Online). http://sipi.usc.edu/database/database.php?volume=misc
26. Pradhan, A.; Sahu, A.K.; Swain, G.; Sekhar, K.R.: Performance evaluation parameters of image steganography techniques. In: IEEE International Conference on Research Advances in Integrated Navigation Systems, pp. 1–8 (2016)
27. El-Emam, N.N.; AL-Zubidy, R.A.S.: New steganography algorithm to conceal a large amount of secret message using hybrid adaptive neural networks with modified adaptive genetic algorithm. J. Syst. Softw. **86**(6), 1465–1481 (2013)
28. El-Emam, N.N.; Al-Diabat, M.: A novel algorithm for colour image steganography using a new intelligent technique based on three phases. Appl. Soft Comput. **37**, 830–846 (2015)
29. El-Emam, N.N.; Qaddoum, K.S.: Improved steganographic security by applying an irregular image segmentation and hybrid adaptive neural networks with modified ant colony optimization. Int. J. Netw. Secur. Appl. **7**(5), 23–47 (2015)
30. Al-Shatanawi, O.M.; El-Emam, N.N.: A new image steganography algorithm based on MLSB method with random pixels selection. Int. J. Netw. Secur. Appl. **7**(2), 37–53 (2015)