Hindawi
Security and Communication Networks
Volume 2017, Article ID 1924618, 13 pages
https://doi.org/10.1155/2017/1924618

Research Article

Adaptive PVD Steganography Using Horizontal, Vertical, and Diagonal Edges in Six-Pixel Blocks

Anita Pradhan, K. Raja Sekhar, and Gandharba Swain

Department of CSE, K L University, Vaddeswaram, Andhra Pradesh 522 502, India

Correspondence should be addressed to Gandharba Swain; gswain1234@gmail.com

Received 29 March 2017; Revised 29 May 2017; Accepted 5 June 2017; Published 2 August 2017

Academic Editor: Ángel Martín Del Rey

The traditional pixel value differencing (PVD) steganographical schemes are easily detected by pixel difference histogram (PDH) analysis. This problem could be addressed by adding two tricks: (i) utilizing horizontal, vertical, and diagonal edges and (ii) using adaptive quantization ranges. This paper presents an adaptive PVD technique using 6-pixel blocks. There are two variants. The proposed adaptive PVD for 2×3-pixel blocks is known as variant 1, and the proposed adaptive PVD for 3×2-pixel blocks is known as variant 2. For every block in variant 1, the four corner pixels are used to hide data bits using the middle column pixels for detecting the horizontal and diagonal edges. Similarly, for every block in variant 2, the four corner pixels are used to hide data bits using the middle row pixels for detecting the vertical and diagonal edges. The quantization ranges are adaptive and are calculated using the correlation of the two middle column/row pixels with the four corner pixels. The technique performs better as compared to the existing adaptive PVD techniques by possessing higher hiding capacity and lesser distortion. Furthermore, it has been proven that the PDH steganalysis and RS steganalysis cannot detect this proposed technique.

1. Introduction

In image steganography techniques, the images are used for covert communication [1]. Since we hide the artificial message inside an image, the image statistics change, so the goal is to minimize this change [2]. Least significant bit (LSB) substitution methods are the age-old methods of image steganography, wherein the LSB bits of a pixel are substituted by secret data bits. This LSB steganography can very easily be detected by RS analysis. Wu and Tsai [3] discovered the fact that the edge regions in an image can conceal more amount of data as compared to the smooth regions. Based on this principle, they proposed pixel value differencing (PVD) steganography. The image should be partitioned into different blocks, each of size of 1×2. For a block, the difference between the two pixels is computed and changed to a new value by hiding data in it. The PVD technique with block size of 2×2 has been proposed to enhance the embedding capacity [4, 5]. Chang and Tseng [6] considered the values of 2, 3, and 4 neighboring pixels to calculate the pixel value differences. But they did not address the FIEP problem, which was resolved in [7]. To achieve higher embedding capacity, Swain [8] proposed PVD steganography considering the maximum difference out of all the differences calculated with all the neighbors. LSB substitution techniques offer higher embedding capacity, but PVD techniques offer higher imperceptibility. Thus, PVD and LSB techniques have been combined to obtain larger hiding capacity and better imperceptibility [9, 10]. Tseng and Leng also proposed one PVD steganography in blocks of size of 1×2 based on the concept of perfect square in the range table [11].

The traditional PVD steganography techniques [3–5] follow a static range table. Due to this, some undesired steps are introduced in pixel difference histograms of the stego images. These step effects can be avoided by applying two tricks: (i) exploiting horizontal, vertical, and diagonal edges and (ii) using adaptive range table. Luo et al. [12] also proposed an adaptive PVD steganography with three-pixel blocks, which does not suffer from the step effects. Swain [13] proposed two PVD steganography techniques using vertical, horizontal, and diagonal edges, which do not suffer from step effects. The first technique uses pixel blocks of size of 2×2 and the second technique uses pixel blocks of size of 3×3. Balasubramanian et al. [14] and Pradhan et al. [15] have proposed PVD

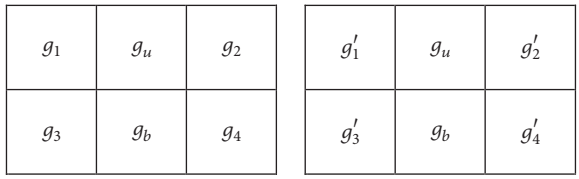


Figure 1: (a) The original block. (b) The stego block.

schemes with 3 × 3 size pixel blocks to achieve higher hiding capacity. Chen [16] proposed a PVD steganography using two reference tables to randomize the data embedding. To improve upon the security, Wang et al. [17] have proposed a different type of PVD steganography. They used modulus function to modify the remainder of a pair of pixels instead of their difference. Lee et al. [18] observed that this technique is also suffering from step effects. So they proposed some improved way to avoid these step effects. Based on pixel value differences, adaptive LSB substitution has been performed in [19, 20]. Nguyen et al. [21] used more than one bit planes and pixel block complexity measure to perform adaptive embedding. In this method, the high textured regions carry more number of bits and low textured regions carry less number of bits. In general, adaptive image steganography schemes possess lower embedding capacity. The edges can be predicted by some prediction functions and hiding capacity depends upon this prediction. Smooth regions cannot hide more number of bits. Based on the level of complexity of the edge regions, adaptive embedding can be applied [22]. Capacity is increased and chance of detection is decreased. To prevent the detection from pixel difference histogram (PDH) analysis, multidirectional edges have been exploited in [15]. A combination of LSB substitution and PVD against RS analysis has been proposed in [23] and a technique based on group of bits substitution against PDH analysis has been proposed in [24]. Compression and encryption techniques can be used with steganography in various ways to add additional measures. In [25], the steganography algorithm uses a public key and a private key to generate pseudorandom numbers that identify the embedding locations.

Being motivated by Luo et al.'s [12] and Swain's [13] adaptive PVD techniques, an adaptive PVD technique with two variants has been proposed using 6-pixel blocks. The number of conditions used for range detection is lesser as compared to Luo et al.'s [12] and Swain's [13] adaptive PVD techniques. The hiding capacity and PSNR have been improved.

2. The Proposed Adaptive PVD Technique with 2 × 3-Pixel Blocks

2.1. The Embedding and Extraction Procedures. The data embedding is performed by traversing the image in raster scan order and splitting the image into blocks of size of 2×3 pixels. A sample block is given in Figure 1(a). The four corner pixels, g_1, g_2, g_3, and g_4, are used to hide confidential message bits, based on the two centre pixels g_u and g_b. The embedding scheme is narrated in the following steps.

Step 1. For $i = 1, 2, 3, 4$, eight difference values, $d_{iu} = (g_i - g_u)$ and $d_{ib} = (g_i - g_b)$, are calculated.

Step 2. For $i = 1, 2, 3, 4$, for target pixel g_i, the lower and upper bounds of the range, l_i and u_i, respectively, are calculated based on the four cases narrated below.

Case 1. If $d_{iu} > 0$ and $d_{ib} > 0$, then $l_i = \max(g_u + 1, g_b + 1)$ and $u_i = 255$.

Case 2. If $d_{iu} \leq 0$ and $d_{ib} \leq 0$, then $l_i = 0$ and $u_i = \min(g_u, g_b)$.

The term $\min(a, b)$ is a function to find the minimum of two values a and b.

Case 3. If $d_{iu} > 0$ and $d_{ib} \leq 0$, then $l_i = g_u + 1$ and $u_i = g_b$.

Case 4. If $d_{iu} \leq 0$ and $d_{ib} > 0$, then $l_i = g_b + 1$ and $u_i = g_u$.

Thus, four sets of lower and upper bounds, (i) l_1 and u_1 for g_1, (ii) l_2 and u_2 for g_2, (iii) l_3 and u_3 for g_3, and (iv) l_4 and u_4 for g_4, are calculated.

Step 3. For $i = 1, 2, 3, 4$, n_i should be the embedding length in g_i. It is estimated by the following equation:

$$n_i = \min\left(\text{floor}\left(\log_2 \left|u_i - l_i + 1\right|\right), 3\right). \tag{1}$$

In (1), floor is a function to find the lower integer of its floating point argument. For example, floor (3.4) = 3.

Step 4. For $i = 1, 2, 3, 4$, n_i bits of secret data are taken and converted to decimal equivalents, b_i. After hiding b_i in g_i, the new value g_i' is calculated as follows:

$$g_i' = \underset{e}{\text{argmin}} \left\{ \left|e - g_i\right| \mid \left|e - g_u\right| \equiv b_i \left(\text{mod } 2^{n_i}\right),\ e \in [l_i, u_i] \right\}. \tag{2}$$

It means, for the original pixel g_i, the stego pixel g_i' is a value e chosen from the range $[l_i, u_i]$ such that it satisfies the following conditions: (i) $(|e - g_u| - b_i) \text{mod } 2^{n_i} = 0$ and (ii) $|e - g_i|$ is minimum. Thus the stego-pixel block of size of 2 × 3 is shown in Figure 1(b).

The data extraction can be performed by traversing the stego image in raster scan fashion and partitioning it into blocks of sizes of 2 × 3. Figure 1(b) is an example of a 2 × 3-stego-pixel block. Data is to be extracted from the pixels g_1', g_2', g_3', and g_4' using the following steps.

Step 1. For $i = 1, 2, 3, 4$, the eight difference values, $d_{iu} = (g_i' - g_u)$ and $d_{ib} = (g_i' - g_b)$, are calculated.

Step 2. For $i = 1, 2, 3, 4$, assume that lower bounds and upper bounds of the ranges for the four corner pixels are l_i and u_i. These are calculated using Step 2 of embedding procedure.

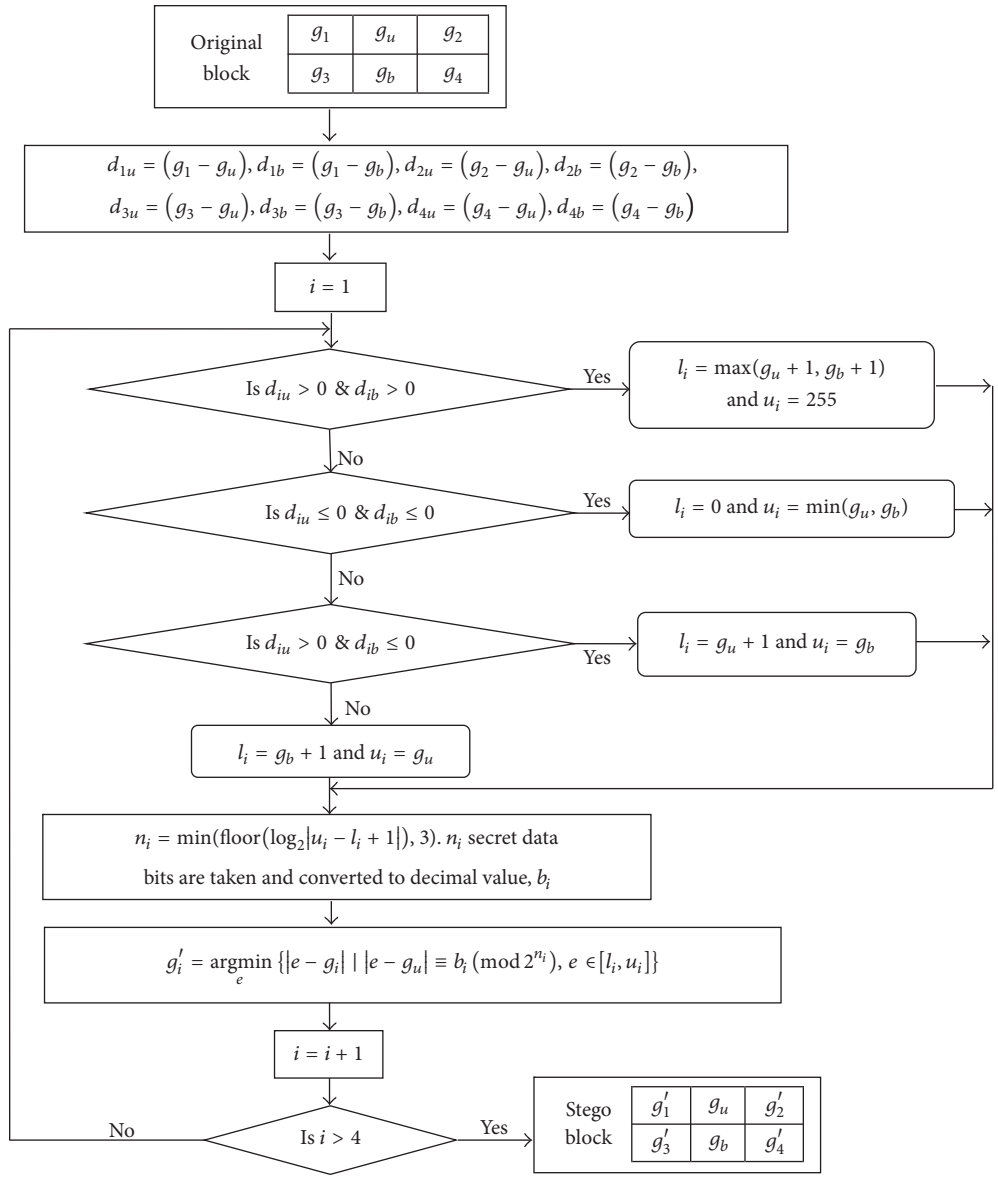


Figure 2: Flow chart for embedding into a 2 × 3-pixel block.

Step 3. The embedding length, n_i, is calculated using the equation in Step 3 of embedding procedure.

Step 4. For $i = 1, 2, 3, 4$, the decimal equivalent of the binary data to be extracted from g_i' is b_i. This is calculated using (3). Finally each b_i is converted to n_i, binary bits.

$$b_i \equiv \left|g_i' - g_u\right| \pmod{2^{n_i}}. \quad (3)$$

Equation (3) means that, out of the list of values of b_i, those satisfying the condition $((b_i - |g_i' - g_u|) \bmod 2^{n_i}) = 0$, the smallest one should be chosen to be the final value of b_i.

2.2. Flow Charts for Embedding and Extraction Procedures. The image is traversed in raster scan order and splitted into nonoverlapped blocks of size of 2 × 3 pixels. The flow chart in Figure 2 represents embedding into a block.

The flow chart in Figure 3 represents the retrieval of secret binary data from a 2 × 3-pixel block.

2.3. Examples of Embedding and Extraction Procedures. An example of embedding is described below. Figure 4(a) is a sample block. Here $g_1 = 98$, $g_u = 100$, $g_2 = 98$, $g_3 = 102$, $g_b = 102$, and $g_4 = 104$.

Step 1. Eight difference values are given as follows:

$$d_{1u} = (g_1 - g_u) = (98 - 100) = -2,$$

$$d_{1b} = (g_1 - g_b) = (98 - 102) = -4,$$

$$d_{2u} = (g_2 - g_u) = (98 - 100) = -2,$$

$$d_{2b} = (g_2 - g_b) = (98 - 102) = -4,$$

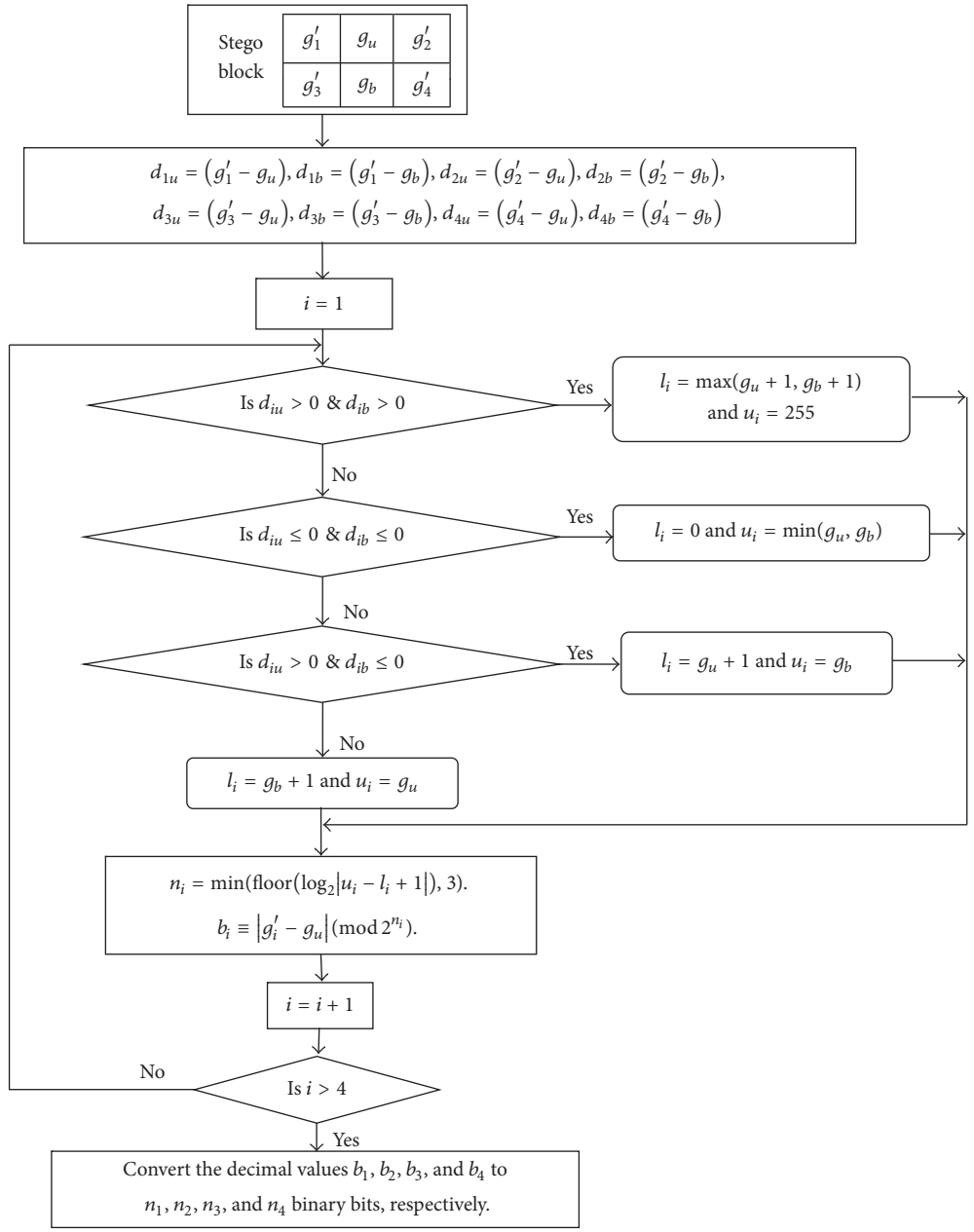


Figure 3: Flow chart for extraction of secret data from a 2 × 3-pixel block.

$$d_{3u} = (g_3 - g_u) = (102 - 100) = 2,$$
$$d_{3b} = (g_3 - g_b) = (102 - 102) = 0,$$
$$d_{4u} = (g_4 - g_u) = (104 - 100) = 4,$$
$$d_{4b} = (g_4 - g_b) = (104 - 102) = 2. \quad (4)$$

Step 2. Now let us find out the lower and upper bounds l_i and u_i for $i = 1, 2, 3$, and 4:

$d_{1u} \le 0$ and $d_{1b} \le 0$. Case 2 is satisfied. Hence, $l_1 = 0$ and $u_1 = \min(g_u, g_b) = 100$.

$d_{2u} \le 0$ and $d_{2b} \le 0$. Case 2 is satisfied. Hence, $l_2 = 0$ and $u_2 = \min(g_u, g_b) = 100$.

$d_{3u} > 0$ and $d_{3b} \le 0$. Case 3 is satisfied. Hence, $l_3 = g_u + 1 = 101$ and $u_3 = g_b = 102$.

$d_{4u} > 0$ and $d_{4b} > 0$. Case 1 is satisfied. Hence, $l_4 = \max(g_u + 1, g_b + 1) = 103$ and $u_4 = 255$.

Step 3.

$$n_1 = \min\left(\text{floor}\left(\log_2\left|u_1 - l_1 + 1\right|\right), 3\right)$$
$$= \min\left(\text{floor}\left(\log_2|100 - 0 + 1|\right), 3\right)$$
$$= \min\left(\text{floor}\left(\log_2|101|\right), 3\right) = \min(6, 3) = 3,$$

$$
\begin{aligned}
n_2 &= \min\left(\text{floor}\left(\log_2\left|u_2 - l_2 + 1\right|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2\left|100 - 0 + 1\right|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2\left|101\right|\right), 3\right) = \min(6, 3) = 3,\\
n_3 &= \min\left(\text{floor}\left(\log_2\left|u_3 - l_3 + 1\right|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2\left|102 - 101 + 1\right|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2\left|102 - 101 + 1\right|\right), 3\right) = \min(1, 3)\\
&= 1,\\
n_4 &= \min\left(\text{floor}\left(\log_2\left|u_4 - l_4 + 1\right|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2\left|255 - 103 + 1\right|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2\left|255 - 103 + 1\right|\right), 3\right) = \min(7, 3)\\
&= 3.
\end{aligned}
\tag{5}
$$

Step 4. Suppose that the secret data stream is $101\,100\,1\,111\,0011\cdots$. Take n_1, that is, 3 bits of data from the secret data stream and convert to its decimal value b_1. Thus, b_1 is 5. Similarly, take n_2, n_3, and n_4 bits of data from the secret data stream and convert to decimal values b_2, b_3, and b_4, respectively. Thus, b_2, b_3, and b_4 values are 4, 1, and 7, respectively.

Now calculate g_1' as follows:

$$
g_1' = \underset{e}{\text{argmin}}\left\{\left|e - g_1\right| \mid \left|e - g_u\right| \equiv b_1 \left(\text{mod } 2^{n_1}\right),\ e \in [l_1, u_1]\right\};
\tag{6}
$$

that is,

$$
g_1' = \underset{e}{\text{argmin}}\left\{|e - 98| \mid |e - 100| \equiv 5 \left(\text{mod } 2^3\right),\ e \in [0, 100]\right\}.
\tag{7}
$$

It means that $(|e - 100| - 5) \bmod 2^3 = 0$ for $e \in [7, 15, 23, 31, 39, 47, 55, 63, 71, 79, 87, 95]$ and g_1' should be chosen as one of the values from this list subject to the condition that $|e - 98|$ is minimum. Thus, g_1' is 95.

Calculate g_2' as follows:

$$
g_2' = \underset{e}{\text{argmin}}\left\{\left|e - g_2\right| \mid \left|e - g_u\right| \equiv b_2 \left(\text{mod } 2^{n_2}\right),\ e \in [l_2, u_2]\right\};
\tag{8}
$$

that is,

$$
g_2' = \underset{e}{\text{argmin}}\left\{|e - 98| \mid |e - 100| \equiv 4 \left(\text{mod } 2^3\right),\ e \in [0, 100]\right\}.
\tag{9}
$$

It means that $(|e - 100| - 4) \bmod 2^3 = 0$ for $e \in [8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96]$ and g_2' should be chosen as one of the values from this list subject to the condition that $|e - 98|$ is minimum. Thus, g_2' is 96.

98	100	98
102	102	104

(a)

95	100	96
101	102	107

(b)

Figure 4: (a) An original sample block and (b) the stego sample block.

Calculate g_3' as follows:

$$
g_3' = \underset{e}{\text{argmin}}\left\{\left|e - g_3\right| \mid \left|e - g_u\right| \equiv b_3 \left(\text{mod } 2^{n_3}\right),\ e \in [l_3, u_3]\right\};
\tag{10}
$$

that is,

$$
g_3' = \underset{e}{\text{argmin}}\left\{|e - 102| \mid |e - 100| \equiv 1 \left(\text{mod } 2^1\right),\ e \in [101, 102]\right\}.
\tag{11}
$$

It means that $(|e - 100| - 1) \bmod 2^1 = 0$ for $e \in [101, 102]$ and g_3' should be chosen as one of the values from this list subject to the condition that $|e - 102|$ is minimum. Thus, g_3' is 101.

Calculate g_4' as follows:

$$
g_4' = \underset{e}{\text{argmin}}\left\{\left|e - g_4\right| \mid \left|e - g_u\right| \equiv b_4 \left(\text{mod } 2^{n_4}\right),\ e \in [l_4, u_4]\right\};
\tag{12}
$$

that is,

$$
g_4' = \underset{e}{\text{argmin}}\left\{|e - 104| \mid |e - 100| \equiv 7 \left(\text{mod } 2^3\right),\ e \in [103, 255]\right\}.
\tag{13}
$$

It means that $(|e - 100| - 7) \bmod 2^3 = 0$ for $e \in [107, 115, 123, 131, \ldots, 227, 235, 243, 251]$ and g_4' should be chosen as one of the values from this list subject to the condition that $|e - 104|$ is minimum. Thus, g_4' is 107.

Thus the stego block is as shown in Figure 4(b).

An example of extraction is as described below. Let us extract the embedded data from the stego block shown in Figure 4(b). Here $g_1' = 95$, $g_u = 100$, $g_2' = 96$, $g_3' = 101$, $g_b = 102$, and $g_4' = 107$.

Step 1. Eight difference values are given as follows:

$$
\begin{aligned}
d_{1u} &= \left(g_1' - g_u\right) = (95 - 100) = -5,\\
d_{1b} &= \left(g_1' - g_b\right) = (95 - 102) = -7,\\
d_{2u} &= \left(g_2' - g_u\right) = (96 - 100) = -4,
\end{aligned}
$$

$$
\begin{aligned}
d_{2b} &= \left(g_2' - g_b\right) = (96 - 102) = -8,\\
d_{3u} &= \left(g_3' - g_u\right) = (101 - 100) = 1,\\
d_{3b} &= \left(g_3' - g_b\right) = (101 - 102) = -1,\\
d_{4u} &= \left(g_4' - g_u\right) = (107 - 100) = 7,\\
d_{4b} &= \left(g_4' - g_b\right) = (107 - 102) = 5.
\end{aligned} \tag{14}
$$

Step 2. Now let us find out the lower and upper bounds l_i and u_i for $i = 1, 2, 3$, and 4:

$d_{1u} \le 0$ and $d_{1b} \le 0$. Case 2 is satisfied. Hence, $l_1 = 0$ and $u_1 = \min(g_u, g_b) = 100$.

$d_{2u} \le 0$ and $d_{2b} \le 0$. Case 2 is satisfied. Hence, $l_2 = 0$ and $u_2 = \min(g_u, g_b) = 100$.

$d_{3u} > 0$ and $d_{3b} \le 0$. Case 3 is satisfied. Hence, $l_3 = g_u + 1 = 101$ and $u_3 = g_b = 102$.

$d_{4u} > 0$ and $d_{4b} > 0$. Case 1 is satisfied. Hence, $l_4 = \max(g_u + 1, g_b + 1) = 103$ and $u_4 = 255$.

Step 3.

$$
\begin{aligned}
n_1 &= \min\left(\text{floor}\left(\log_2\left|u_1 - l_1 + 1\right|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2 |100 - 0 + 1|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2 |101|\right), 3\right) = \min(6, 3) = 3,\\
n_2 &= \min\left(\text{floor}\left(\log_2\left|u_2 - l_2 + 1\right|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2 |100 - 0 + 1|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2 |101|\right), 3\right) = \min(6, 3) = 3,\\
n_3 &= \min\left(\text{floor}\left(\log_2\left|u_3 - l_3 + 1\right|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2 |102 - 101 + 1|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2 |102 - 101 + 1|\right), 3\right)\\
&= \min(1, 3) = 1,\\
n_4 &= \min\left(\text{floor}\left(\log_2\left|u_4 - l_4 + 1\right|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2 |255 - 103 + 1|\right), 3\right)\\
&= \min\left(\text{floor}\left(\log_2 |255 - 103 + 1|\right), 3\right)\\
&= \min(7, 3) = 3.
\end{aligned} \tag{15}
$$

Step 4. For $i = 1, 2, 3, 4$, the decimal equivalent of the binary data to be extracted from g_i' is b_i. That means b_1 is extracted from g_1', b_2 is extracted from g_2', b_3 is extracted from g_3', and b_4 is extracted from g_4'. It is performed as follows.

$$
b_1 \equiv \left|g_1' - g_u\right| \pmod{2^{n_1}}; \tag{16}
$$

that is,

$$
b_1 \equiv |95 - 100| \pmod{2^3}. \tag{17}
$$

It means that $(b_1 - |95 - 100|) \bmod 2^3 = 0$.

$(b_1 - 5) \bmod 8 = 0$ implies that minimum value of b_1 is 5. This b_1 is converted to n_1 binary bits. That means 5 is converted to 3 binary bits as 101_2.

$$
b_2 \equiv \left|g_2' - g_u\right| \pmod{2^{n_2}}; \tag{18}
$$

that is,

$$
b_2 \equiv |96 - 100| \pmod{2^3}. \tag{19}
$$

It means that $(b_2 - |96 - 100|) \bmod 2^3 = 0$.

$(b_2 - 4) \bmod 8 = 0$ implies that minimum value of b_2 is 4. This b_2 is converted to n_2 binary bits. That means 4 is converted to 3 binary bits as 100_2.

$$
b_3 \equiv \left|g_3' - g_u\right| \pmod{2^{n_3}}; \tag{20}
$$

that is,

$$
b_3 \equiv |101 - 100| \pmod{2^3}. \tag{21}
$$

It means that $(b_3 - |101 - 100|) \bmod 2^3 = 0$.

$(b_3 - 1) \bmod 8 = 0$ implies that minimum value of b_3 is 1. This b_3 is converted to n_3 binary bits. That means 1 is converted to a single binary bit. It is 1_2.

$$
b_4 \equiv \left|g_4' - g_u\right| \pmod{2^{n_4}}; \tag{22}
$$

that is,

$$
b_4 \equiv |107 - 100| \pmod{2^3}. \tag{23}
$$

It means that $(b_4 - |107 - 100|) \bmod 2^3 = 0$.

$(b_4 - 7) \bmod 8 = 0$ implies that minimum value of b_4 is 7. This b_4 is converted to n_4 binary bits. That means 7 is converted to 3 binary bits as 111_2.

Thus the extracted bits are 101 100 1 111. These are the embedded bits.

3. The Proposed Adaptive PVD Technique with 3 × 2-Pixel Blocks

The data embedding is performed by traversing the image in raster scan order and splitting the image into blocks of size of 3×2 pixels. A sample block is given in Figure 5(a). The four corner pixels, g_1, g_2, g_3, and g_4, are used to hide confidential message bits, based on the two centre pixels g_L and g_r. The embedding scheme is narrated in the steps below.

Step 1. For $i = 1, 2, 3, 4$, eight difference values, $d_{iL} = (g_i - g_L)$ and $d_{ir} = (g_i - g_r)$, are calculated.

Step 2. For $i = 1, 2, 3, 4$, for target pixel g_i, the lower and upper bounds of the range, l_i and u_i, respectively, are calculated based on the four cases narrated below.

Case 1. If $d_{iL} > 0$ and $d_{ir} > 0$, then $l_i = \max(g_L + 1, g_r + 1)$ and $u_i = 255$.

Case 2. If $d_{iL} \le 0$ and $d_{ir} \le 0$, then $l_i = 0$ and $u_i = \min(g_L, g_r)$.

g_1	g_2
g_L	g_r
g_3	g_4

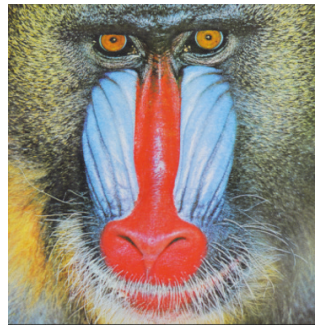
(a)

g'_1	g'_2
g_L	g_r
g'_3	g'_4

(b)

FIGURE 5: (a) The original block. (b) The stego block.

(a) Lena



(b) Baboon

(c) Peppers

(d) Jet

FIGURE 6: The original images.

Case 3. If $d_{iL} > 0$ and $d_{ir} \leq 0$, then $l_i = g_L + 1$ and $u_i = g_r$.

Case 4. If $d_{iL} \leq 0$ and $d_{ir} > 0$, then $l_i = g_r + 1$ and $u_i = g_L$.

Thus four sets of lower and upper bounds, (i) l_1 and u_1 for g_1, (ii) l_2 and u_2 for g_2, (iii) l_3 and u_3 for g_3, and (iv) l_4 and u_4 for g_4, are calculated.

Step 3. For $i = 1, 2, 3, 4$, the embedding length in g_i is n_i. It is estimated by (24). The term $\min(a, b)$ is a function to find the minimum of two values a and b.

$$n_i = \min\left(\text{floor}\left(\log_2 \left|u_i - l_i + 1\right|\right), 3\right). \tag{24}$$

Step 4. For $i = 1, 2, 3, 4$, the n_i bits of secret data are taken and converted to decimal equivalents, b_i. After hiding b_i in g_i, the new value g'_i is calculated as follows:

$$g'_i = \underset{e}{\text{argmin}} \left\{\left|e - g_i\right| \mid \left|e - g_L\right| \equiv b_i \left(\text{mod } 2^{n_i}\right), \; e \in [l_i, u_i]\right\}. \tag{25}$$

Thus the stego-pixel block of size of 3×2 is as given in Figure 5(b).

The data extraction can be performed by traversing the stego image in raster scan fashion and partitioning into blocks of sizes of 3×2. Figure 5(b) is an example of a 3×2-stego-pixel block. Data is to be extracted from the pixels g'_1, g'_2, g'_3, and g'_4 using the following steps.

Step 1. For $i = 1, 2, 3, 4$, the eight difference values, $d_{iL} = (g'_i - g_L)$ and $d_{ir} = (g'_i - g_r)$, are calculated.

Step 2. For $i = 1, 2, 3, 4$, assume that lower bounds and upper bounds of the ranges for the four corner pixels are l_i and u_i. These are calculated using Step 2 of embedding procedure.

Step 3. The embedding length, n_i, is calculated using the equation in Step 3 of the embedding procedure.

Step 4. For $i = 1, 2, 3, 4$, the decimal equivalent of the binary data to be extracted from g'_i is b_i. This is calculated using (26). Finally, each b_i is converted to n_i binary bits.

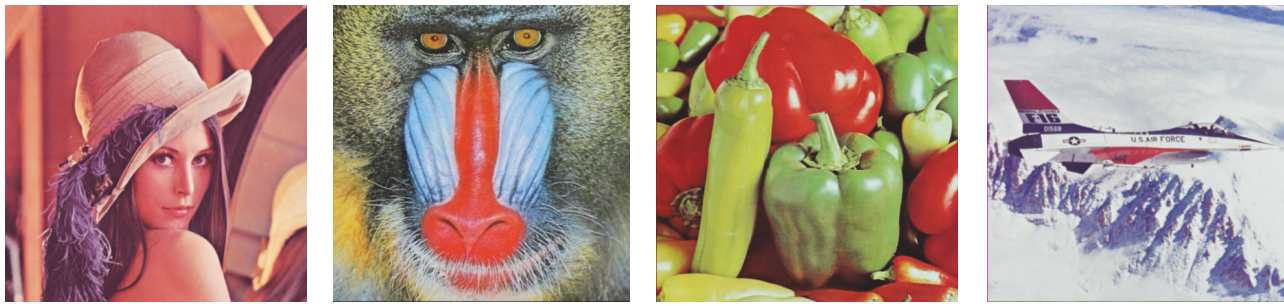
$$b_i \equiv \left|g'_i - g_L\right| \quad \left(\text{mod } 2^{n_i}\right). \tag{26}$$

4. Experimental Results and Discussion

MATLAB is used for simulating this technique. The tested images are collected from SIPI database. Figure 6 represents some original sample images. Figures 7 and 8 show their respective stego images. The embedding length in each of these stego images is 140000 (one lakh and forty thousand) bits. The stego images are as good as the original images.

(a) (b) (c) (d)

Figure 7: The stego images for adaptive PVD technique in 2 × 3-pixel blocks (variant 1).



(a) (b) (c) (d)

Figure 8: The stego images for adaptive PVD technique in 3 × 2-pixel blocks (variant 2).

The proposed adaptive PVD schemes are compared with the adaptive PVD techniques in [12, 13]. There are three parameters considered for comparison. The first parameter is peak signal-to-noise ratio (PSNR). It is an estimation of distortion in the stego image. Higher value of PSNR implies lesser amount of distortion. The PSNR is calculated in terms of mean square error (MSE). Equations (27) and (28) represent the calculations for MSE and PSNR, respectively:

$$\text{MSE} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(p_{ij} - q_{ij}\right)^2, \tag{27}$$

$$\text{PSNR} = 10 \times \log_{10} \frac{255 \times 255}{\text{MSE}}. \tag{28}$$

The second parameter is hiding capacity. It is the maximum amount of information that can be concealed in the stego image, represented in bits. It can also be represented as bits per byte (BPB). This BPB is often termed as bit rate; it is calculated as in (29). It is used to express the hiding capacity per a byte of the image [13]:

$$\text{BPB} = \frac{\text{Maximum hiding capacity in bits}}{\text{Image size in bytes}}. \tag{29}$$

The third parameter, quality index (Q), is a measure of equivalence between two images. It is evaluated by (30). If the two images are exactly the same, then quality index is 1.

$$Q = \frac{4\sigma_{xy}\overline{p}\,\overline{q}}{\left(\sigma_x^2 + \sigma_y^2\right)\left[\left(\overline{p}\right)^2 + \left(\overline{q}\right)^2\right]}. \tag{30}$$

The terms used in (28) are as follows. The terms $\overline{p}$ and $\overline{q}$ are the mean pixel value of the original image and stego image, respectively. The terms σ_x^2 and σ_y^2 are the standard deviation for the original image and stego image, respectively. The term σ_{xy} is the covariance. The estimation of all these symbols is done using (31), (32), (33), (34), and (35), respectively:

$$\overline{p} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}, \tag{31}$$

$$\overline{q} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} q_{ij}, \tag{32}$$

$$\sigma_x^2 = \frac{1}{m \times n - 1} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(p_{ij} - \overline{p}\right)^2, \tag{33}$$

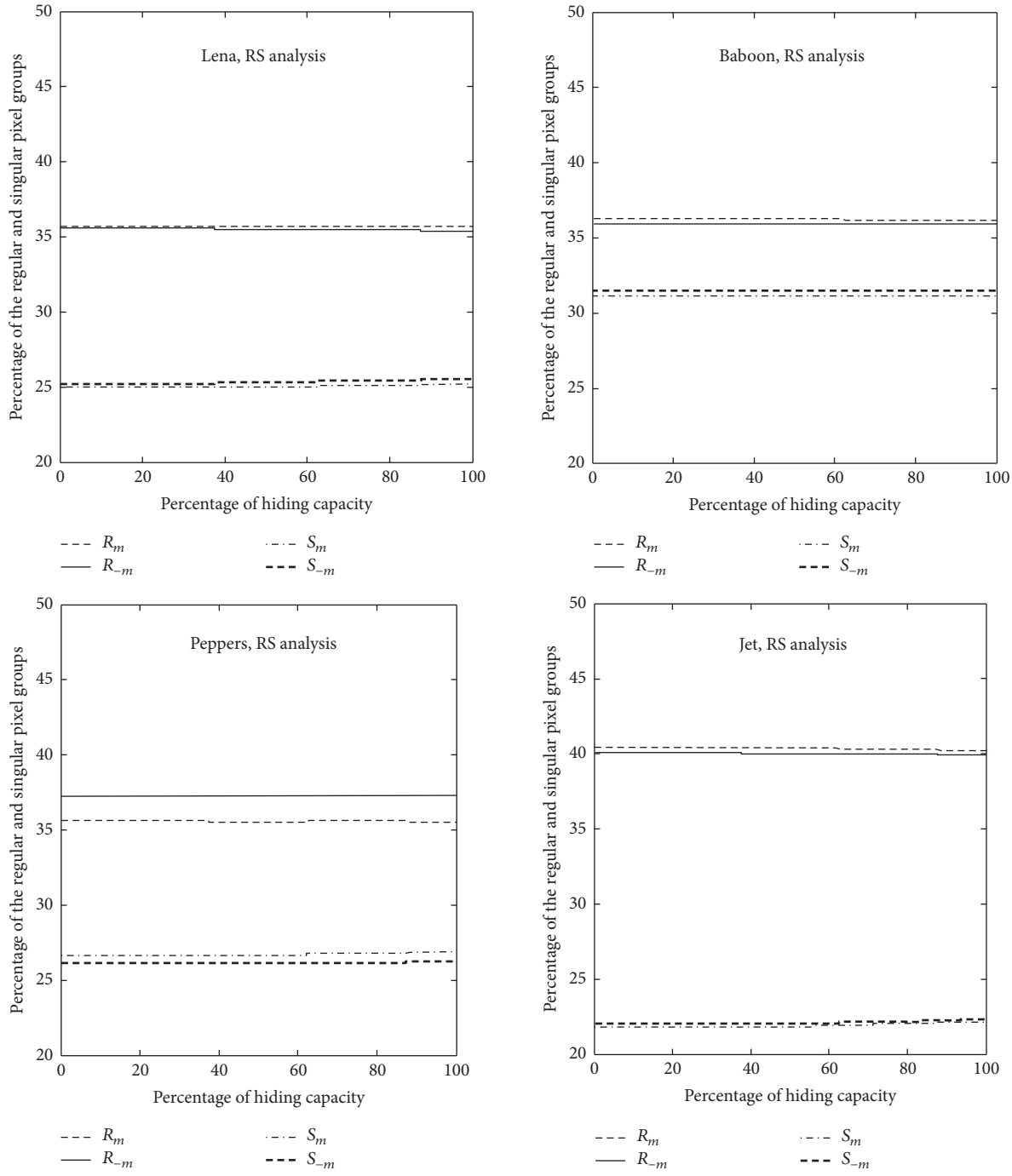


Figure 9: RS analysis of proposed variant 1 for four Images.

$$\sigma_y^2 = \frac{1}{m \times n - 1} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(q_{ij} - \overline{q}\right)^2, \tag{34}$$

$$\sigma_{xy} = \frac{1}{m \times n - 1} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(p_{ij} - \overline{p}\right)\left(q_{ij} - \overline{q}\right). \tag{35}$$

Looking at the last rows in Tables 1 and 2, the average over the results of seven sample images is given. It can be noticed that both the capacity and PSNR of the proposed techniques are improved over the existing techniques proposed in [12, 13].

The security of our proposed technique is evaluated by tools like (i) RS steganalysis and (ii) PDH steganalysis. The RS steganalysis graphs of Lena, Baboon, Peppers, and Jet images are represented in Figures 9 and 10 for proposed variant 1 and variant 2, respectively. In all the cases, graphs for R_m and R_{-m} are linear and very near to each other. Similarly, the graphs

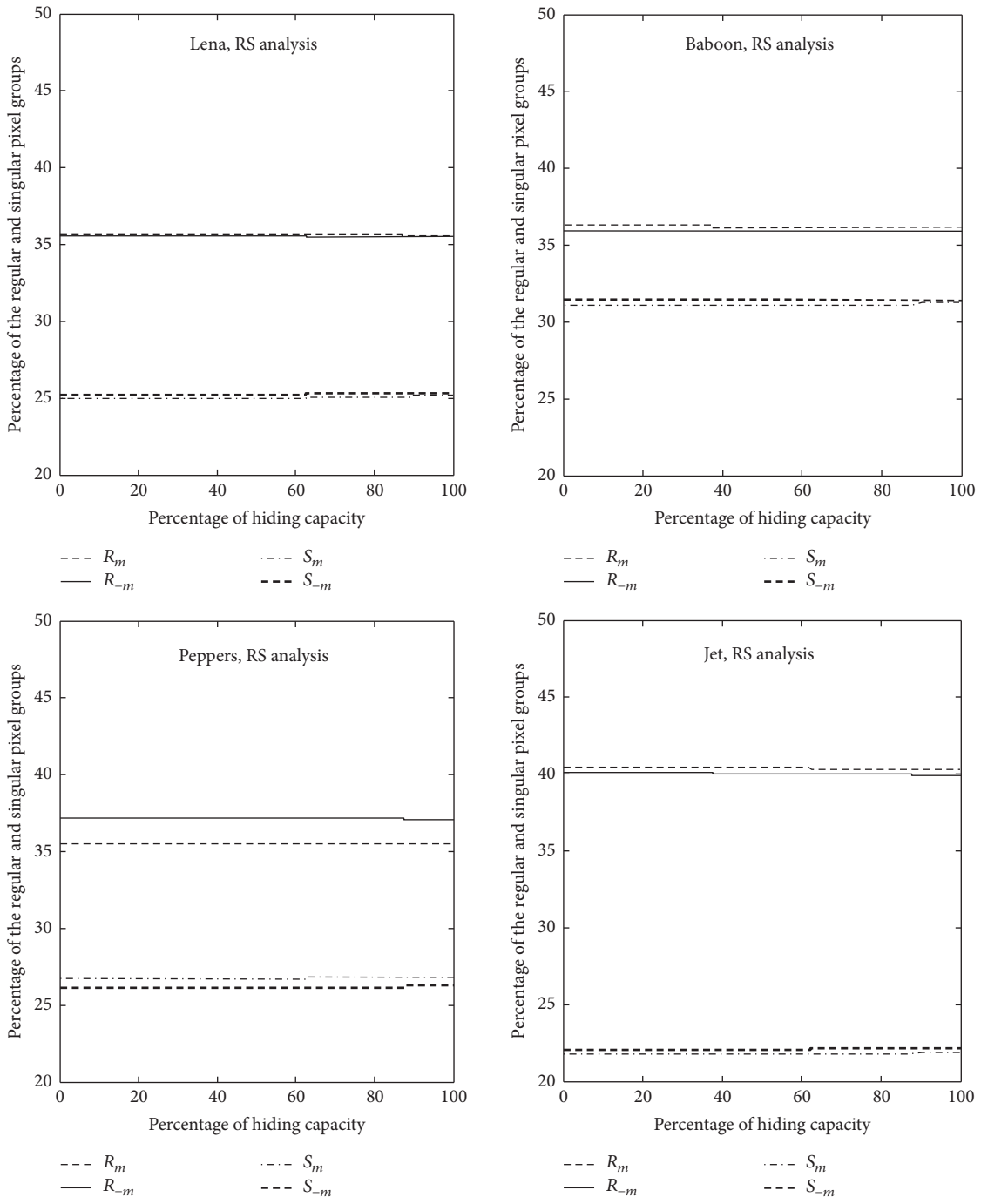


Figure 10: RS analysis of proposed variant 2 for four Images.

for S_m and S_{-m} are linear and close to each other. Hence, the relation $R_m \cong R_{-m} > S_m \cong S_{-m}$ is valid. So we can conclude that the proposed technique cannot be detected by RS analysis.

Figures 11 and 12 represent the PDH analysis for Lena, Baboon, Peppers, and Jet images for proposed variant 1 and variant 2, respectively. In all these eight graphs, the solid line curve is for original image and the dotted line curve is for stego image. We can verify that there are no step effects in the curves of the stego images. This proves that the technique could not be detected by pixel difference histogram analysis.

5. Conclusion

The PVD steganography techniques are well known for higher imperceptibility. But for the traditional PVD steganography techniques, the pixel difference histograms show some step effects. This problem has been addressed by using two

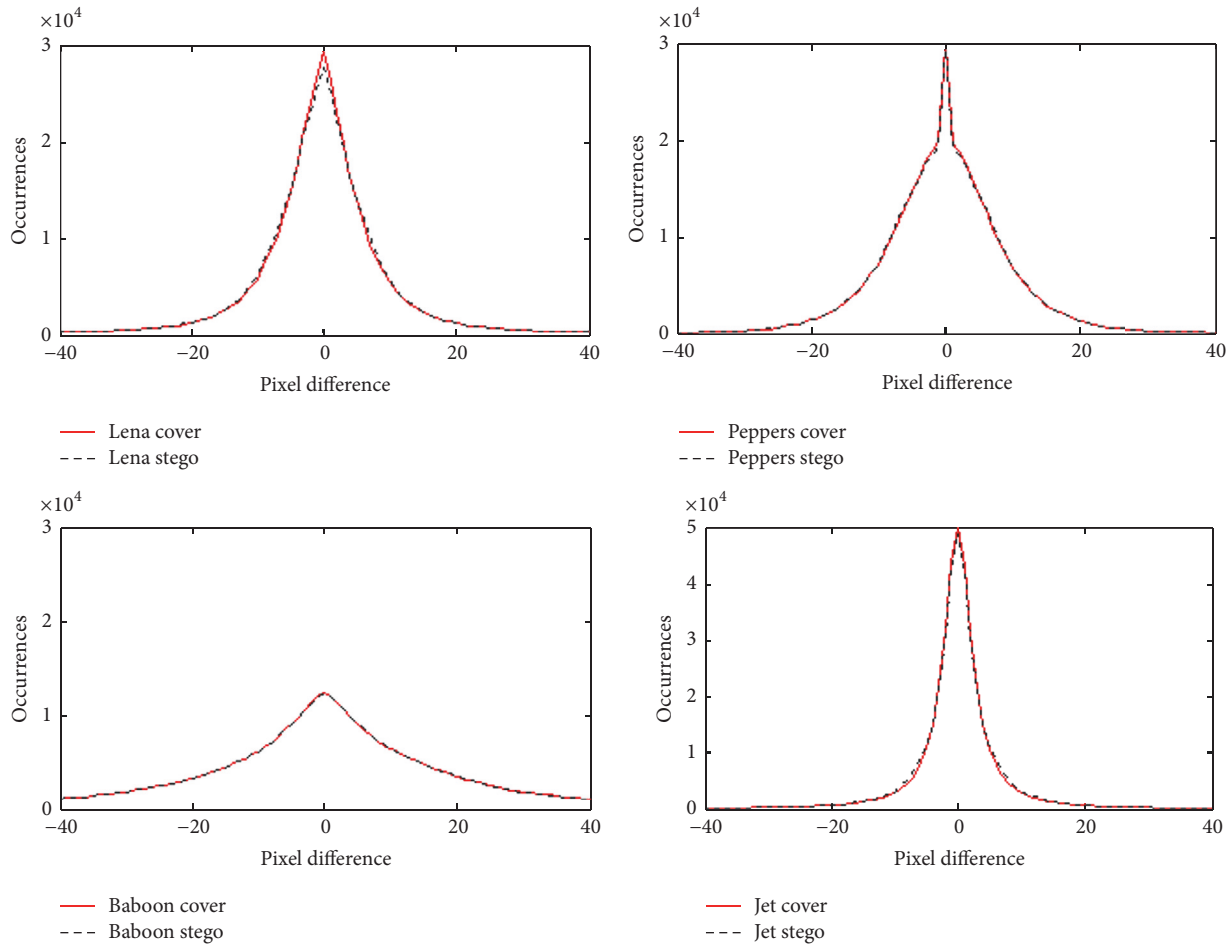


Figure 11: PDH analysis for four images (variant 1).

Table 1: Results of existing techniques.

Images 512 × 512	Swain's 2 × 2-pixel block adaptive PVD [13]				Luo et al.'s 1 × 3-pixel block adaptive PVD [12]			
	PSNR	Capacity	Q	BPB	PSNR	Capacity	Q	BPB
Lena	45.04	1341191	0.999	1.70	48.79	229037	0.999	0.29
Baboon	47.13	1489945	0.999	1.89	48.03	611197	0.999	0.77
Peppers	45.73	1350251	0.999	1.71	48.32	264058	0.999	0.33
Jet	44.86	1267690	0.999	1.61	48.76	145755	0.999	0.18
Boat	46.08	1424967	0.999	1.81	48.20	389588	0.999	0.49
House	43.58	1339985	0.999	1.70	48.41	259413	0.999	0.32
Tiffany	45.23	1341498	0.999	1.70	48.70	165873	0.999	0.21
Average	45.37	1365075	0.999	1.61	48.45	294988	0.999	0.37

tricks: (i) exploiting vertical, horizontal, and diagonal edges and (ii) using adaptive quantization ranges. In both proposed variants, the four corner pixels are used to hide data bits and the two middle row/column pixels are used to detect the edges. The quantization ranges are adaptive and are calculated using correlation of the middle pixels with the four corner pixels. From the observations, it is evident that this proposed technique achieves higher embedding length and lesser distortion as compared to that of Luo et al.'s and Swain's adaptive PVD techniques. Furthermore, it has been

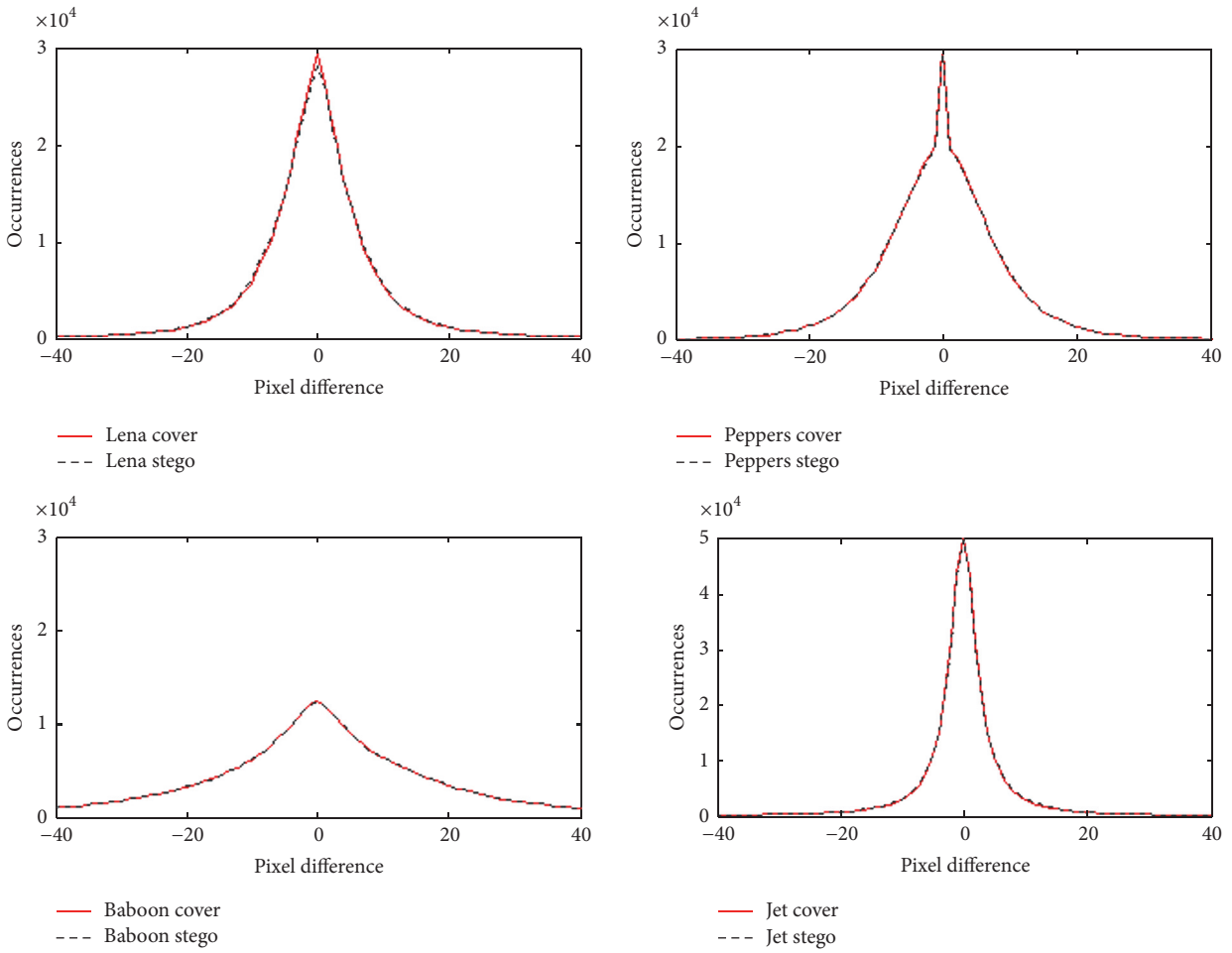


Figure 12: PDH analysis for four images (variant 2).

Table 2: Results of proposed technique.

Images 512 × 512	Proposed 2 × 3-pixel block adaptive PVD				Proposed 3 × 2-pixel block adaptive PVD			
	PSNR	Capacity	Q	BPB	PSNR	Capacity	Q	BPB
Lena	50.89	1445784	0.999	1.83	50.61	1425521	0.999	1.81
Baboon	52.29	1532417	0.999	1.94	52.36	1527208	0.999	1.94
Peppers	51.29	1418101	0.999	1.80	51.22	1409621	0.999	1.79
Jet	50.65	1381432	0.999	1.75	50.77	1362765	0.999	1.73
Boat	51.42	1479835	0.999	1.88	51.43	1474106	0.999	1.88
House	49.09	1431346	0.999	1.82	49.18	1429845	0.999	1.82
Tiffany	50.88	1430606	0.999	1.81	50.95	1420300	0.999	1.81
Average	**50.93**	**1445645**	0.999	1.83	**50.93**	**1435623**	0.999	1.82

experimentally proven that the RS steganalysis and pixel difference histogram steganalysis cannot detect this proposed technique.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. J. Anderson and F. A. P. Petitcolas, "On the limits of steganography," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 474–481, 1998.

[2] A. Martín, G. Sapiro, and G. Seroussi, "Is image stenography natural?" *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2040–2050, 2005.

[3] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1613–1626, 2003.

[4] K. C. Chang, C. P. Chang, P. S. Huang, and T. M. Tu, "A novel image steganographic method using tri-way pixel-value differencing," *Journal of Multimedia*, vol. 3, no. 2, pp. 37–44, 2008.

[5] Y.-P. Lee, J.-C. Lee, W.-K. Chen, K.-C. Chang, I.-J. Su, and C.-P. Chang, "High-payload image hiding with quality recovery using tri-way pixel-value differencing," *Information Sciences*, vol. 191, pp. 214–225, 2012.

[6] C.-C. Chang and H.-W. Tseng, "A steganographic method for digital images using side match," *Pattern Recognition Letters*, vol. 25, no. 12, pp. 1431–1437, 2004.

[7] G. Swain and S. K. Lenka, "Steganography using two sided, three sided, and four sided side match methods," *CSI Transactions on ICT*, vol. 1, no. 2, pp. 127–133, 2013.

[8] G. Swain, "Steganography in digital images using maximum difference of neighboring pixel values," *International Journal of Security and its Applications*, vol. 7, no. 6, pp. 285–294, 2013.

[9] H. C. Wu, N. I. Wu, C. S. Tsai, and M.-S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," *IEE Proceedings—Vision, Image and Signal Processing*, vol. 152, no. 5, pp. 611–615, 2005.

[10] C.-H. Yang, C.-Y. Weng, S.-J. Wang, and H.-M. Sun, "Varied PVD + LSB evading detection programs to spatial domain in data embedding systems," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1635–1643, 2010.

[11] H.-W. Tseng and H.-S. Leng, "A steganographic method based on pixel-value differencing and the perfect square number," *Journal of Applied Mathematics*, vol. 2013, Article ID 189706, 2013.

[12] W. Luo, F. Huang, and J. Huang, "A more secure steganography based on adaptive pixel-value differencing scheme," *Multimedia Tools and Applications*, vol. 52, no. 2-3, pp. 407–430, 2011.

[13] G. Swain, "Adaptive pixel value differencing steganography using both vertical and horizontal edges," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13541–13556, 2016.

[14] C. Balasubramanian, S. Selvakumar, and S. Geetha, "High payload image steganography with reduced distortion using octonary pixel pairing scheme," *Multimedia Tools and Applications*, vol. 73, no. 3, pp. 2223–2245, 2014.

[15] A. Pradhan, K. R. Sekhar, and G. Swain, "Digital image steganography based on seven way pixel value differencing," *Indian Journal of Science and Technology*, vol. 9, no. 37, pp. 11-10, 2016.

[16] J. Chen, "A PVD-based data hiding method with histogram preserving using pixel pair matching," *Signal Processing: Image Communication*, vol. 29, no. 3, pp. 375–384, 2014.

[17] C.-M. Wang, N.-I. Wu, C.-S. Tsai, and M.-S. Hwang, "A high quality steganographic method with pixel-value differencing and modulus function," *Journal of Systems and Software*, vol. 81, no. 1, pp. 150–158, 2008.

[18] H.-K. Lee, J.-C. Joo, and H.-Y. Lee, "Improved steganographic method preserving pixel-value differencing histogram with modulus function," *Eurasip Journal on Advances in Signal Processing*, vol. 2010, Article ID 249826, 2010.

[19] M. Khodaei and K. Faez, "New adaptive steganographic method using least-significant-bit substitution and pixel-value differencing," *IET Image Processing*, vol. 6, no. 6, pp. 677–686, 2012.

[20] X. Liao, Q. Wen, and J. Zhang, "A steganographic method for digital images with four-pixel differencing and modified LSB substitution," *Journal of Visual Communication and Image Representation*, vol. 22, no. 1, pp. 1–8, 2011.

[21] T. D. Nguyen, S. Arch-Int, and N. Arch-Int, "An adaptive multi bit-plane image steganography using block data-hiding," *Multimedia Tools and Applications*, vol. 75, no. 14, pp. 8319–8345, 2016.

[22] S. Chakraborty, A. S. Jalal, and C. Bhatnagar, "LSB based non blind predictive edge adaptive image steganography," *Multimedia Tools and Applications*, pp. 1–15, 2016.

[23] G. Swain, "A steganographic method combining LSB substitution and PVD in a block," *Procedia Computer Science*, vol. 85, pp. 39–44, 2016.

[24] G. Swain, "Digital image steganography using variable length group of bits substitution," *Procedia Computer Science*, vol. 85, pp. 31–38, 2016.

[25] A. Soria-Lorente and S. Berres, "A Secure Steganographic Algorithm Based on Frequency Domain for the Transmission of Hidden Information," *Security and Communication Networks*, vol. 2017, pp. 1–14, 2017.