

A Modification of the TPVD Algorithm for Data Embedding

J.A. Hernández-Servin^(✉), J. Raymundo Marcial-Romero,
Vianney Muñoz Jiménez, and H.A. Montes-Venegas

Facultad de Ingeniería, Universidad Autónoma Del Estado de México, CU,
Cerro Coatepec, Toluca, Edo México, México
{xoseahernandez, jrmarcialr, vmunozj, hamontesv}@uaemex.mx

Abstract. Pixel-Value-Differencing (PVD) methods for data hiding have the advantage of a high payload. These algorithms have however the problem of overflow/underflow pixels, thus a location map for those pixels usually ignored when embedding the message is necessary. In this paper, we modified the Tri-way Pixel-Value Differencing method that removes the need of the location map and fix the problem. Our proposal replaces the table of ranges to estimate the amount of information to be embedded by a function based on the floor and ceil functions. As for the problem of overflow/underflow pixels we tackle it by means of a linear transformation. The linear transformation is based on the floor function, so information is lost therefore a location map to compensate for this data lost is necessary to recover the embedded message. The inclusion of the map in the algorithm is also discussed. The technique uses two steganographic methods, namely, the tri-way method to store the message and a reversible steganographic method to store the map needed to invert the linear function in order to recover the encoded message.

Keywords: Steganography · Tri-way pixel-value differencing · TPVD · Reversible · Data hiding

1 Introduction

The technique for embedding useful information into digital covers such as images is called data hiding or steganography. The medium to be used as a cover is not restricted to images, it can be any form of digital signal. In this paper, 8-bits digital images are taken as digital covers. There is a characteristic that distinguishes steganography as opposed to cryptography in that not necessarily the message has to be encrypted but must be an innocuous-looking object. In general terms, a good steganographic algorithm for data hiding must have: (1) high embedding rate measured by bits per pixel (bpp), (2) distortion of the host must be low; the distortion is normally measured by the peak signal to noise ratio (PSNR) in dB units and (3) the method has to be immune to steganalysis, as much as possible, in the sense that a hacker must not even suspect there is a hidden message in an image. The third characteristic is specially difficult to comply. One could

say, that there are three categories of methods for doing steganography that can loosely be classified as pixel-value-differencing (PVD) [1, 2], least-significant-bit (LSB) [3] and methods based on linear transformations such as Fourier [4] and Wavelet transforms [5, 6]. Even though there is a growing interest among *Coding Theorist* for developing new steganographic techniques from a code theory point of view [7]. Many methods had been subject to steganalysis and LSB methods, as originally published, are the weakest to such analysis [8]. The PVD methods have been also analysed by steganalysis, some authors [9] claim that PVD methods and their derivations generate abnormal high fluctuations in PVD histograms making this particular method prone to detection. Despite these drawbacks there is still high interests in PVD methods and their improvement for the high payload they provide.

Many steganographic methods have the problem that when embedding a message some pixels of the stego image exceed the 8-bit range $[0, 255]$. This problem constrain the algorithm to ignore those pixels that fall off boundaries for message insertion. Additionally, the decoder has to know the exact location of pixels that are used for insertion and the location of the pixels being ignored.

In this paper, we propose a simple modification to the tri-way pixel-value differencing (TPVD) method [2] to avoid building a location map of the pixels ignored for embedding. We are also dealing with the problem of overflow/underflow pixels by proposing a simple linear transformation that potentially increase the payload while keeping a reasonable peak signal to noise ratio. In fixing overflow/underflow pixels we ran into additional difficulties as we also need a map to revert the process in order to recover the embedded message. To deal with this problem, we propose to use the resultant stego image, that is the image after message insertion, to embed the map using a reversible data hiding method. This simple idea has been explored elsewhere for different purposes [10]. Reversible data hiding is the subject of intense research where the purpose is not only to recover the embedded message but also to recover the host image. This problem has been tackled with a wavelet approach as reported in [5, 6].

The paper is organised as follows. In Sect. 2, we briefly review the TPVD method, present our proposal and introduce some necessary notation for the rest of the paper. Section 3 presents the method to fix overflow/underflow pixels of the TPVD method. Finally, in Sect. 4 we show that our technique really works by analysing some images.

2 A Modified TPVD Method for Data Hiding

This section explains in detail the *Tri-way Pixel-Value Differencing* (TPVD) [2] method for steganographic data embedding. Let us assume that M is an image with pixel value range in $J = [0, 2^8 - 1] \cap \mathbb{N}$. It will be understood that M is a matrix in $J^{m \times n}$ which can be partitioned in blocks of size 2×2 . That is, $M = \{[B_{uv}]\}_{u,v}$ with $[B_{uv}] \in J^{2 \times 2}$. For every block $[B]$ in M we define the distance block matrix as $[d] \in J^{2 \times 2}$ where $[d]_{ij} = [B]_{ij} - [B]_{11}$. In the TPVD method a set of ranges are defined in order to decide the amount of information

to be embedded in every block. The set of ranges $R = \{[l, u] | l \leq u; l, u \in J\}$ are supposed to be fix and shall be shared to the decoder. Analogously, to the definition of block distance, we can define the lower and upper block for every block $[B]$ in M . Thus, $[l]_{ij}, [u]_{ij}$ are those intervals in R such that $[l]_{ij} \leq |[d]_{ij}| \leq [u]_{ij}$. Likewise, it is defined the block $[w] = [u] - [l] + 1$ and $[t] = \lfloor \log_2([w]) \rfloor$ where the \log_2 is performed on each entry of the block $[w]$. Each entry in $[t]$ is the amount of bits to be embedded on each pixel of block $[B]$ from the binary message. The message is normally plain text where each character is converted into its decimal equivalent following the ascii table standard then into its binary counterpart.

2.1 Our Proposal

Instead of considering a set of ranges we propose to embed $[t]$ bits from message where each entry is defined as

$$[t]_{ij} = \begin{cases} 0 & i, j = 1 \\ \phi(|[d]_{ij}|) & \text{otherwise} \end{cases} \quad (1)$$

where $\phi(z) = \lceil \log_2(z + 2H(1 - z)) \rceil$ and H is the heaviside function defined as $H(z) = 1$ if $z \geq 0$ and 0 otherwise. If we assume the message to embed is given as a string b of $0's$ and $1's$, for instance $b = 001010101010 \dots$, then we can take blocks of size $\sum_{i,j} [t]_{ij}$ then we build the message block

$$[b]_{ij} = \sum_{s=0}^{[t]_{ij}-1} b_{ijs} 2^{[t]_{ij}-s-1} \quad (2)$$

where the b_{ijs} are the bits that compose the message b . For example, taken an arbitrary b we have that

$$b = \dots \underbrace{0 \dots 1}_{[t]_{12}} \underbrace{0 \dots 0}_{[t]_{21}} \underbrace{1 \dots 1}_{[t]_{22}} \dots \quad (3)$$

To embed the message let us consider the block matrix $2^{[t]}$ where each entry is defined as $2^{[t]_{ij}}$ for every i, j . Define a new block difference $[d']$ on each entry i, j as

$$[d']_{ij} = (2 \cdot H([d]_{ij}) - 1)(2^{[t]_{ij}} + [b]_{ij}). \quad (4)$$

Three more blocks will be defined in order to obtain the stego image. The difference between $[d]$ and $[d']$ is denoted by $[m]$ that is $[m] = [d] - [d']$ and based on $[m]$ we define the auxiliary blocks $[\alpha], [\beta]$ as

$$\alpha_{ij} = \left\lceil \frac{m_{ij}}{2} \right\rceil \cdot H(m_{ij}) + \left\lfloor \frac{m_{ij}}{2} \right\rfloor H(-m_{ij}) \quad (5)$$

and

$$\beta_{ij} = \left\lceil \frac{m_{ij}}{2} \right\rceil \cdot H(-m_{ij}) + \left\lfloor \frac{m_{ij}}{2} \right\rfloor H(m_{ij}) \quad (6)$$

and finally define the block matrix Φ

$$[\Phi(p, q)]_{ij} = \begin{cases} -[\alpha]_{pq} & i, j = 1 \\ [\alpha]_{ij} + [\beta]_{ij} - [\alpha]_{pq} & \text{otherwise} \end{cases} \quad (7)$$

Therefore the stego block $[B'(p, q)]$ is given by $[B'(p, q)] = [B] - [\Phi(p, q)]$

Optimal Stego-block. The mean square error for any pair of matrices $M, N \in J^{s \times t}$ is defined as

$$|M - N| = \frac{1}{st} \sum_i^s \sum_j^t (M_{ij} - N_{ij})^2. \quad (8)$$

Thus, the optimal stego-block can be chosen as

$$[B'] = \min_{\substack{p \neq 1 \\ q \neq 1}} \{ |[\Phi(p, q)]| \} \quad (9)$$

2.2 Decoding

To decode, that is to obtain the embedded message from the stego-block $[B']$ we define the distance block matrix as $[d^*] = [B'] - \mathbf{I} \cdot [B']_{11}$ where \mathbf{I} is the matrix with $\mathbf{I}_{ij} = 1$ for all i, j . The block $[t^*]$ is calculated replacing $[d]$ by $[d^*]$ in Eq. (1). Therefore the message is recovered by

$$[b] = \left(\frac{[d^*]}{2H([d^*]) - 1} \right) \bmod 2^{[t^*]} \quad (10)$$

where the function $m \bmod n$ denotes the residual part obtained after dividing the integer m over n . One of the features of the PVD methods is that the differences are kept unmodified which can readily be verified.

Proposition 1. *The differences between neighbour pixels before and after embedding are the same. That is, $[d'] = [d^*]$ for every block in which the host image is being partitioned.*

Proof. Straightforward.

Claim. By defining the amount of bits to be embedded through the function ϕ as in Eq. 1 we avoid the problem of having a map of exact locations of the ignored pixels in order to fully recover the message.

Proof. By Proposition 1 above we have that for all i, j except when $i = j = 1$

$$\lfloor \log_2 |[d^*]_{ij} | \rfloor = \left\lfloor \log_2 \left(2^{[t]_{ij}} + [b]_{ij} \right) \right\rfloor = [t]_{ij} \quad (11)$$

Thus, from Eq. 11, the block $[t]$ gives a clear indication of which blocks are being ignored. When $[t] = 0$ the coder ignores that particular block.

3 Correcting Stego-blocks

In general any steganographic PVD method has the problem that when inserting a message some pixel might fall off the valid boundary. That is, the stego-block $[B']$ might not be in $J^{2 \times 2}$. To correct it, we are proposing a simple linear transformation that results into a valid stego-block. For this, let us assume that $[B']$ is in $I^{2 \times 2}$ where $I = [c^-, c^+] \cap \mathbb{N}$ is a finite interval of integers such that $c^- = \min I \leq 0$ and $255 \leq c^+ = \max I$. Let us define $C = \frac{J_1 - J_0}{c^+ - c^-}$, $D = \frac{J_0 c^+ - J_1 c^-}{c^+ - c^-}$ then the transformation proposed is given by $\psi : I \rightarrow J$, $\psi(x) = Cx + D$ and obviously its inverse is given by $\psi^{-1} : J \rightarrow I$, $\psi^{-1}(y) = \frac{y}{C} - \frac{D}{C}$.

The simplicity of the function does give an equally simple scheme to produce a valid stego image, we can simply define $S = \lfloor \psi(B') \rfloor \in J^{2 \times 2}$. However, to recover the message is not that straightforward. The function ψ is clearly bijective, and it can be checked that $J = \psi^{-1}(\psi(J))$ but $J \neq \psi(J)$ since $\psi(J)$ are not integers but real numbers. However, $\psi(J) \in [0, 255] \subset \mathbb{R}$ that is $J_0 = \min \psi(J)$ and $J_1 = \max \psi(J)$. In other words, the function ψ is evenly embedding J into J . We can naively say that $\psi^{-1}(S)$ recovers B' , however ψ^{-1} is not a function formally speaking but an inverse relation.

In order to fully recover B' , we must be able to build the inverse relation both accurately and efficiently. As S is the image that the decoder must have, thus we must be able to recover B' from it. Let us consider $S = \lfloor \psi(I) \rfloor$, and denote the inverse relation by $\Xi = (\xi, \tau)$ such that $\Xi(s) = x$ for every $x \in I$ and $s \in S$. We will show that Ξ in fact has the following form $x = \Xi(s) = \min \xi_s + \tau(s)$ with $\tau \in \{0, 1\}$ and ξ_s is a set of integer in I that depends on s . Let us define the set of indices that depend on C as $K = \{0, \dots, 2 \lceil \frac{1}{C} \rceil + 1\}$.

Claim. For every $s \in S$ such that $s = \lfloor \psi(x) \rfloor$ for some $x \in I$; let us consider the sequence $s_k = \psi \lceil \psi^{-1}(s) \rceil + Ck$ for all $k \in K$ then if $A_s = \{k \in K | s \leq s_k \leq s + 1\}$ and $\lceil \frac{1}{C} \rceil \leq 2$

- (i) the set ξ_s is given by $\xi_s = \{\lceil \psi^{-1}(s) \rceil + k\}_{k \in A_s}$,
- (ii) and $\tau(s) = x - \min \xi_s$ defines the map τ .

Proof. Straightforward.

The map τ is a binary matrix of same dimension as the stego image, that is all entries in τ are 0's and 1's. This only happens when $|I| \leq 2|J|$ or equivalently when $c^- \leq 126$ and $c^+ \leq 255 + 126$. Beyond that point the inverse relation $\Xi(s) \geq 2$ and the map needs more than two integers to encode the inverse relation. This obviously, although possible, will make the stego image file bigger as it requires more space.

3.1 Encoding the Map

Once the stego image S is obtained from the modified TPVD scheme, the image S is, in general, out of the proper pixel value range. By applying a correction, as explained in Sect. 3, we end up with a valid image M and a map τ . The decoder must know the map τ somehow. We propose to use a reversible steganographic

method to embed the map τ . Here, the image to be used as host is again the stego image M . This simple idea was first explored by [10]. They, however, used a different method for embedding the message.

A reversible steganographic scheme is a method where the embedded message as well as the host image are fully recovered. The first method with such characteristics was proposed by Tian [6]; later improvements and generalisations were carried out by [5]. There are some other methodologies proposed for reversible data hiding, for a brief review see for instance [11]. The problem with reversible techniques is that they usually need a map to reverse the process. However, the authors do not discuss the problem of where to embed the map to let the decoder extract the message.

In this paper, we are proposing to use the steganographic scheme by [5] or simply use the proposed by [6]. The scheme in [6] is based on the well known Haar wavelet. This technique has also the problem that some pixels may fall off the valid boundaries when embedding data. Those pixels must be ignored since they cannot be used for embedding. However, the decoder must know the exact location of the ignored pixels. The important part of the algorithm by [5,6] is that they efficiently managed to embed the map of the ignored pixels as part of the message.

4 Experimental Work

The experimental work was carried out on 512×512 standard images as shown in Fig. 1. The level of distortion is measured using the standard peak signal to noise ratio (PSNR) and embedding capacity is measured in bits per pixel (bpp). We choose those images in order to compare our results with [5]. The authors in [5] do compare their findings with some other proposals, so we think is an interesting comparison.

The algorithm proposed in this paper has the flexibility to ignore overflow/underflow pixels, as other PVD methods, but without the need of marking off the ignored pixels. We run two sets of experiments on five images, the first one ignores overflow/underflow pixels and the second one allows overflow/underflow pixels and then fixes them according to Sect. 3.

4.1 Ignoring Overflow/Underflow Pixels

Table 1(a) shows the results of embedding data at its maximum capacity for the five images shown in Fig. 1. We are reproducing the values reported in [5] for the same set of images for the comparison.

We also carried out experiments by restricting the maximum value for $[t]$ to be less than four. That is, we allow the algorithm to embed each pixel difference at most four bits of message.

At first glance, by comparing results from Table 1(a) and (b) we observe an unexpected and counter-intuitive behaviour. That is, less data embedded should result in less distortion of the host image which is not observed by comparing the PSNR values obtained in the two experiments. It seems that this is normal

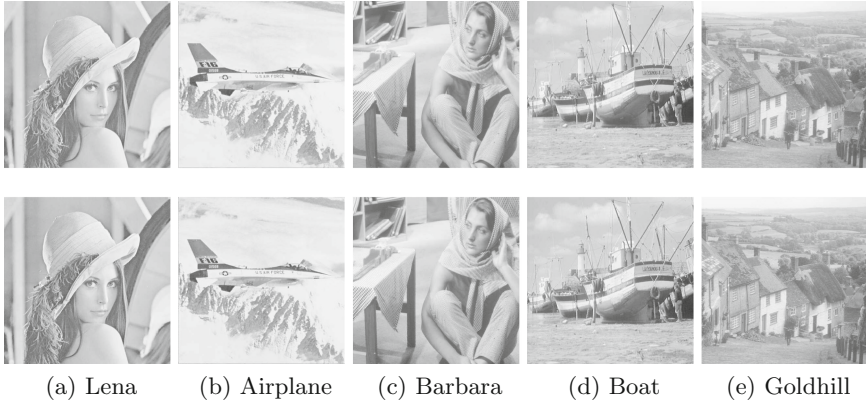


Fig. 1. Original images (first row), and the images obtained after data embedding (second row) using our modified TPVD.

behaviour in the TPVD method. Restricting the maximum value for the $[t]$ blocks to less than seven, the algorithm behaves in such a way that introduces some pepper noise in the edge regions of the stego image; as a result, the PSNR goes below the optimum (Fig. 2).

The amount of bpp for each image in Table 1(a), is calculated according to the following formula, $\text{bpp} = \frac{1}{mn} \sum_{[t] \in M} \sum_{\substack{i=1 \\ j=1}}^2 [t]_{ij}$ where M denotes the host image seen as a matrix of dimension $m \times n$. In the implementation, the entry $[t]_{11} = 0$ since the pivot pixel is not used for embedding, thus the entry does not contribute to the sum, even so it is always modified as it can be seen from Eq. 4.

Figure 3 shows the effect on fixing overflow/underflow pixels in the Boat image.

Table 1. (a) Comparison of the results obtained by ignoring overflow/underflow pixels against results published by [5] using the Haar wavelet technique. This experiment is run allowing the maximum in t to be ≤ 7 for every block (b) Comparison of the results obtained by ignoring pixels out of range against results published by [5]. The maximum bits permitted for embedding at each pixel difference is four, that is $\max t \leq 4$.

	Our proposal		F. Peng [5]	
	PSNR	bpp	PSNR	bpp
Lena	36.04	1.38	30.75	1.2
Airplane	36.09	1.3	33.45	1.2
Barbara	34.56	1.8	26.66	1.2
Boat	37.03	1.66	30.7	1.2
Goldhill	37.55	1.6	26.89	1.2

(a)

	Our proposal		F. Peng [5]	
	PSNR	bpp	PSNR	bpp
Lena	32.53	1.36	30.75	1.2
Airplane	28.72	1.25	33.45	1.2
Barbara	25.22	1.68	26.66	1.2
Boat	30.29	1.61	30.7	1.2
Goldhill	31.68	1.57	26.89	1.2

(b)

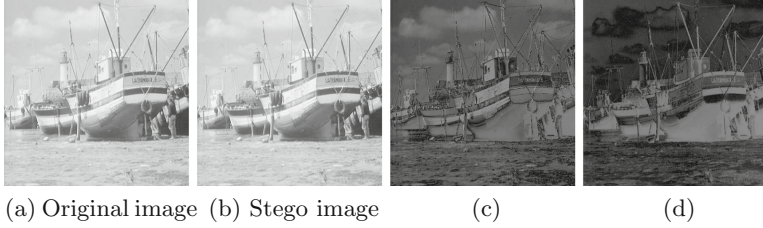


Fig. 2. The boat image shows the method explained in Sect. 3 to fix the overflow/underflow pixels common in PVD methods. Images shown in (c), (d) are the difference between host image and stego image with $\max[t] \leq 4$ and $\max[t] \leq 7$ respectively.

4.2 Comparison Against TPVD

The function ϕ in Eq. 1, proposed as a modification to the range table of the original TPVD scheme is twofold. We can either define ϕ using $\lfloor \cdot \rfloor$ or $\lceil \cdot \rceil$, floor and ceiling functions respectively. The differences, although subtle, have some impact on the message payload.

Figure 3 shows the empirical probability distribution for both the Goldhill and stego image for comparison. It can be seen some similarities, however, there are some shifts in the histogram that have some impact on the PSNR value.

By comparing the values between Tables 2 and 3b, it can clearly be seen that the choice of the ceiling rather than the floor function makes a difference on the message payload. It does increase the payload but the PSNR, in some cases, gets not desirable values. We can see that if we keep $\max[t] \leq 4$ we can substantially increase the payload without compromising the distortion of the host image. However, in the original paper [2], the authors reported that for the Lena image they managed to embed 75,836 bytes which roughly correspond to a $\frac{(75836)(8)}{512^2} \approx 2.3$ bpp in an image of size 512×512 and PSNR of 38.8. This is much higher than expected. By comparing Table 3a and b we can clearly see that by choosing the ceiling function and ignoring the overflow/underflow pixels, we can improve the PSNR values and get higher payload, even so, they are not that

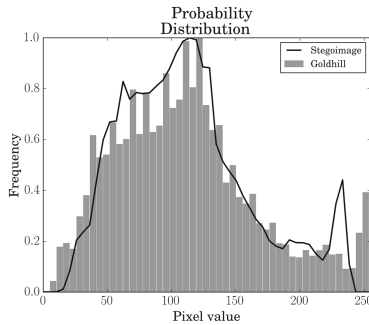


Fig. 3. Emprirical dsitribution of the stegoimage and Goldhill image.

Table 2. Experimental results where pixels are allowed to fall off boundaries, that is $c^- = -126$ and $c^+ = 126$. In this instance the function ϕ is defined by means of the floor function $\lfloor \cdot \rfloor$.

	$\max[t] \leq 4$		$\max[t] \leq 7$	
	PSNR	bpp	PSNR	bpp
Lena	22.12	1.36	23.58	1.38
Airplane	24.7	1.25	21.03	1.3
Barbara	21.42	1.69	22.17	1.81
Boat	20.21	1.62	22.06	1.67
Goldhill	31.68	1.57	37.05	1.6

Table 3. (a) The values shown in this table were performed with $c^- = c^+ = 0$ and the function ϕ is defined by means of the ceiling function $\lceil \cdot \rceil$. (b) The values shown in this table were performed with $c^- = -126$ and $c^+ = 126$ and the function ϕ is defined by means of the ceiling function $\lceil \cdot \rceil$.

	$\max[t] \leq 4$		$\max[t] \leq 7$			$\max[t] \leq 4$		$\max[t] \leq 7$	
	PSNR	bpp	PSNR	bpp		PSNR	bpp	PSNR	bpp
Lena	30.31	1.77	27.4	1.79	Lena	21.95	1.77	20.43	1.85
Airplane	28.02	1.64	27.34	1.63	Airplane	24.6	1.64	17.24	1.75
Barbara	24.85	2.04	24.19	1.98	Barbara	21.35	2.05	18.91	2.33
Boat	28.48	2.04	25.65	2.05	Boat	20.18	2.04	20.49	2.18
Goldhill	29.83	2.0	25.78	2.05	Goldhill	28.6	2.0	19.76	2.12

(a)

(b)

higher as those reported by [2]. We must, however, emphasize that the choice of ϕ instead of the range table removes the need of a map for the exact location of the ignored pixels. Also, it is worth noting that the authors in [2] do not discuss how the decoder knows which pixels or blocks are being ignored. We think is an important part of the problem of steganographic methods.

5 Conclusions

In this paper, we proposed a modified algorithm as an alternative to the TPVD scheme. The modification has the advantage that when ignoring overflow/underflow pixels the decoder does not need a map to know the exact location of the ignored pixels as it is encoded in the function ϕ proposed. The proposal of Sect. 3 is independent of the choice of the function ϕ for the amount of information to embedded. In other words, we can use the original table of ranges in order to keep the same embedded rate as reported in [2], while keeping the flexibility of fixing those overflow/underflow pixels. Although not reported in this paper, we carried out some experiments that supports the thesis that, by fixing overflow/underflow pixels, the payload does notably increase while keeping a reasonable PSNR value. These still remains to be reported and further investigated.

Although our results did not improve the original TPVD scheme, it still performs much better compared to others methods reported in the literature. Also, we have highlighted some fundamental problems that are not studied or discussed in depth in most PVD methods. We are developing a methodology to tackle those problems but this is still an ongoing research.

References

1. Wu, D.-C., Tsai, W.-H.: A steganographic method for images by pixel-value differencing. *Pattern Recogn. Lett.* **24**(9–10), 1613–1626 (2003). doi:[10.1016/S0167-8655\(02\)00402-6](https://doi.org/10.1016/S0167-8655(02)00402-6)
2. Changa, K.-C., Changa, C.-P., Huangb, P.S., Tu, T.-M.: A novel image steganographic method using tri-way pixel-value differencing. *J. Multimed.* **3**(2), 37–44 (2008)
3. Qazanfari, K., Safabakhsh, R.: A new steganography method which preserves histogram: generalization of lsb++. *Inf. Sci.* **277**, 90–101 (2014). doi:[10.1016/j.ins.2014.02.007](https://doi.org/10.1016/j.ins.2014.02.007)
4. Rekik, S., Guerchi, D., Selouani, S.-A., Hamam, H.: Speech steganography using wavelet and fourier transforms. *EURASIP J. Audio, Speech, Music Process.* **2012**, 1123–1125 (2012)
5. Peng, F., Li, X., Yang, B.: Adaptive reversible data hiding scheme based on integer transform. *Signal Process.* **92**, 54–62 (2012)
6. Tian, J.: Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **13**(8), 890–896 (2003)
7. Zhang, W., Li, S.: Steganographic codes—a new problem of coding theory. *J. Latex Class Files* **1**(11), 1–7 (2002)
8. Fridrich, J., Goljan, M., Høgea, D.: Steganalysis of jpeg images: breaking the f5 algorithm. In: Petitcolas, F.A.P. (ed.) *IH 2002*. LNCS, vol. 2578, pp. 310–323. Springer, Heidelberg (2003)
9. Bui, C.-N., Lee, H.-Y., Joo, J.-C., Lee, H.-K.: Steganalysis method defeating the modified pixel-value differencing steganography. *Int. J. Innovative Comput. Inf. Control* **6**(7), 3193–3203 (2010)
10. Liu, G., Dai, Y., Wang, Z., Shiguo, L.: Adaptive image steganography by reversible data hiding. In: Li, T., Xu, Y., Ruan, D. (Eds.) *International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2007)* (2007)
11. Sarkar, T., Sanyal, S.: Reversible and irreversible data hiding technique, CoRR [abs/1405.2684](https://arxiv.org/abs/1405.2684). <http://arxiv.org/abs/1405.2684>