Computers, Materials & Continua
DOI:10.32604/cmc.2021.016605
Article

Tech Science Press

A Reversible Data Hiding Algorithm Based on Image Camouflage and Bit-Plane Compression

Jianyi Liu[1], Ru Zhang[1,*], Jing Li[2], Lei Guan[3], Cheng Jie[2] and Jiaping Gui[4]

[1]School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[2]Network Security Monitoring Center, State Grid Information & Telecommunication Branch, Beijing, 100761, China
[3]School of Electrical Engineering, Tsinghua University, Beijing, 100091, China
[4]Data Science and System Security Department, NEC Labs America, Princeton, 08540, America
[*]Corresponding Author: Ru Zhang. Email: zhangru@bupt.edu.cn
Received: 06 January 2021; Accepted: 06 February 2021

Abstract: Reversible data hiding in encrypted image (RDHEI) is a widely used technique for privacy protection, which has been developed in many applications that require high confidentiality, authentication and integrity. Proposed RDHEI methods do not allow high embedding rate while ensuring losslessly recover the original image. Moreover, the ciphertext form of encrypted image in RDHEI framework is easy to cause the attention of attackers. This paper proposes a reversible data hiding algorithm based on image camouflage encryption and bit plane compression. A camouflage encryption algorithm is used to transform a secret image into another meaningful target image, which can cover both secret image and encryption behavior based on "plaintext to plaintext" transformation. An edge optimization method based on prediction algorithm is designed to improve the image camouflage encryption quality. The reversible data hiding based bit-plane level compression, which can improve the redundancy of the bit plane by Gray coding, is used to embed watermark in the camouflage image. The experimental results also show the superior performance of the method in terms of embedding capacity and image quality.

Keywords: Reversible data hiding; image camouflage; bit plane compression; encryption; edge optimization

1 Introduction

Reversible data hiding (RDH) plays a significant role in data hiding field, and RDH scheme has demonstrated its strong potential in different applications. In many application scenarios, however, it is desirable to carry out RDH scheme directly on encrypted images. Such an approach is called reversible data hiding in encrypted images (RDHEI). RDHEI can embed secret information into encrypted images, with a reversible manner that the original covers can be losslessly decrypted and recovered after the embedded information are extracted. RDHEI can find many applications, e.g., for military communications, medical systems and cloud storage.

The RDHEI mainly includes three-party roles: content owner, information hider, and receiver. Some RDHEI schemes are vacating room after encryption (VRAE) schemes, which create redundancy after encryption process. VRAE schemes have the disadvantage that the embedding rate is low and errors may be occurred during image reconstructing phase. On the other hand, there are reserving room before encryption (RRBE) schemes, which reserve redundancy space before encryption. Although RRBE can achieve a high embedding rate, it might be impractical as the content owner needs to do an extra workload to create the space for the information hider.

For both VRAE and RRBE, the content owner will make encryption process on the secret image and send the encrypted image to the information hider. Although encryption can protect the secret image in a certain extent, but the messy codes of the encrypted image are easy to cause the attention of attackers who may try to dig out information on the content owner. In this paper, we propose a novel RDHEI schemes based on image camouflage and bit-plane compression. A camouflage encryption algorithm is used to transfer the original secret image into another meaningful image. The "plaintext to plaintext" transformation can cover both secret image and encryption behavior. Because the encrypted camouflage image is still a meaningful image that has different semantic content from original secret image, it will avoid the attention of the attacker while protecting the original secret image. Compared to exist image camouflage, an edge optimization method based on prediction algorithm is designed to improve the image encryption quality. The reversible data hiding based bit-plane level compression, which can improve the redundancy of the bit plane by Gray coding, is used to embed watermark in the camouflage image.

2 Related Works

The RRBE scheme preserves the embedded space by extracting compressible features in the plaintext domain of the carrier image. It requires the content owner to perform additional operations to obtain the embedded space before encryption without any embedded information. The reserved space is often a relatively concentrated contiguous areas, which may cause copyright protection loopholes. However, due to its real conveniency, high efficiency, and good embedding performance, RRBE is suitable for some applications that have not strict security requirements.

Mo et al. [1] propose a novel RDHEI scheme based on block classification and permutation, which firstly divides the original image into smooth and non-smooth blocks, uses stream encryption and block permutation to protect image, and then embeds secret information in the most significant bits (MSB) of the encrypted pixels in smooth blocks. Wang et al. [2] propose a specific steganalysis scheme for histogram-shifting based reversible data hiding method, which provides an effective framework associated with 'flat ground' detection and double check modules to improve detection accuracy. Ma et al. [3] use the existing RDH to embed the Least Significant Bit (LSB) of some pixels of the carrier into the other pixels to create the embedded space. After encryption, the reversible embedding of secret data is replaced by LSB, which can reach the embedding rate of 0.5bpp (bit per pixel). Similarly, Yi et al. [4] propose a binary block embedding algorithm that embeds the binary bits in the lower bit plane into the higher bit plane to reserve the embedding space, which achieves excellent performance under fully reversible conditions. Zhang et al. [5] divide the plaintext image into sample pixels and non-sample pixels, and the corresponding prediction error of sample pixels is obtained by the prediction algorithm. Then, they use the security encryption algorithm and the specific encryption mode to process the non-sample pixels and the prediction error to obtain the ciphertext carrier. Using the histogram shift to preserve the predicted error of the prediction, the reversible effect is considerable, but the embedding rate is not ideal (less than 0.1 bpp). Shiu et al. [6] use the differential expansion algorithm

to reserve the embedded space in advance, and then use the Paillier homomorphic encryption algorithm to encrypt. After embedding the selected pixel pairs into the secret information, the secret information and the original carrier can be recovered by the homomorphic characteristics without distortion. Cao et al. [7] propose an encrypted image reversible steganography algorithm based on patch-level sparse coding technology, which uses an over-complete dictionary for sparse representation of image blocks and makes full use of pixel correlation compression to obtain a large embedded space.

The VRAE scheme is completely carried out on the encrypted domain, so it is not easy to leak plaintext information. It has a higher practical value and attracts more attention. Zhang [8] firstly proposes the idea of combining image encryption and information hiding, and designs the first reversible data hiding algorithm in the ciphertext domain. The algorithm implements the embedding of secret information by flipping the last three LSBs of a specific pixel of the ciphertext carrier. The receiver uses the spatial correlation to extract the secret information and restore the original image, which satisfies certain reversibility. But this method must decrypt firstly and then extract the information. Considering the optimization of block smoothness and the correlation of boundary pixels, Hong et al. [9] propose an improved smooth-evaluation function combined with a side matching scheme to reduce the error rate of information extraction, which offers a better performance of the algorithm to some extent. However, information extraction errors are still inevitable. Flip ratio and wave prediction function are used to optimize the algorithm [10–14], which reduce the error of extracting bits and recovering images. But the effective embedding rate is lower than 0.75bpp, and it is always accompanied by an incomplete reversible state. Liu et al. [15] design an encryption algorithm to maintain the entropy of the carrier. The embedding scheme combined with bit-plane sparse matrix compression achieves complete reversibility and obtains better algorithm performance, but the security of the scheme is poor due to the leakage of carrier statistics. From the perspective of information theory, Karim et al. [16] propose and prove that the entropy of random encrypted signal tends to but does not reach the maximum value. Using GRC entropy encoding for encrypted signals can obtain redundancy for information embedding, but the residual entropy space is too small. The maximum embedding rate is maintained at 0.169 bpb (bit per bit).

RDH technique based on homomorphic encryption has the advantages of higher embedding rate as well as better visual performance. Wu et al. [17] use the homomorphism of Shamir secret sharing to retain data differences, and implement reversible steganography in the ciphertext domain by DE and HS, achieving high embedding rate and low computational complexity while being completely reversible. Zhang et al. [18] and Xiang et al. [19] successively use homomorphic encryption algorithms LWE and Paillier to preserve data correlation, and further design a special RDH algorithm to improve security and embedding performance. Zhang et al. [20] propose a RDH scheme which combines lossless and reversible methods by combining Paillier, Damgård-Jurik, and multi-layer wet paper coding, which can simultaneously perform two types of embedding operations. However, the above algorithms are difficult to avoid the ciphertext expansion problem, which results in storage space expansion and communication bandwidth increase. Li et al. [21] and Xiao et al. [22] combine with DE and the additive homomorphic encryption algorithm proposed by the references [23] to achieve reversible steganography in ciphertext domain. This algorithm does not produce encryption extension, and at the same time, it well conceals the statistical information of the carrier. Huang et al. [24] and Yi et al. [25] propose a ciphertext reversible steganography algorithm framework based on sub-block pixel-level xor encryption and

block scrambling, which can be directly combined with the RDH in the plaintext domain to achieve the reversible steganography in the encrypted domain.

Specifically, Lai et al. [26] propose "secret-fragment-visible" mosaic image, which is created automatically by selecting the target image similar to secret image in a database and replacing the similar blocks between secret blocks and target blocks, achieving an effect of embedding the given image visibly but secretly in the resulting mosaic image. The method is only suitable for a target image similar with the secret image, and the camouflage image quality is not so good. Lee et al. [27] proposed an approximately reversible block conversion method based on sub-block variance to improve the transformation effect. Although Lee et al.'s method can transform a secret image to a freely-selected target image without a database, it cannot losslessly recover the secret image. Subsequently, Zhang et al. [28] improved the block conversion method into a reversible conversion method, and used the quantile of sub-block standard deviation for classification and matching to improve the image quality. Hou et al. [29,30] made continuous improvement based on the results of Zhang et al. 's method, introduced K-means clustering algorithm to improve the similarity of sub-block matching, obtained good encryption effect. This effect of image camouflage is useful for covert communication or secure keeping of secret images and provides a new research idea for RDHEI. Since the image is still plaintext after camouflage encryption, the RDH algorithm of the plaintext domain can be directly applied.

3 Proposed Reversible Data Hiding Algorithm

Fig. 1 shows the framework of proposed scheme, which can be divided into three phases: (1) carrier encryption phase, (2) information embedding phase, (3) information extraction and secret image recovery phase.

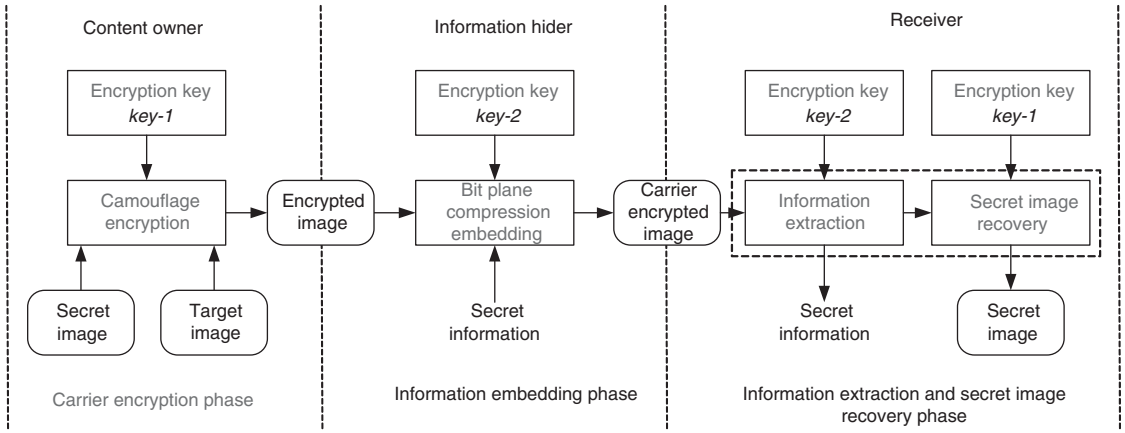

Figure 1: Scheme framework

In the carrier encryption phase, the content owner converts the secret image and the target image in terms of sub-blocks to obtain the transformed camouflage encrypted image with an encryption key *Key-1*. The target image does not need to compare the similarity with the secret image. The camouflage encrypted image is transmitted to the information hider.

In the information embedding phase, the camouflage encrypted image is a meaningful image compared to traditional RDHEI methods. Therefore, the information hider can select a classical

RDH method according to embedding capacity and image quality. This paper uses a RDH based bit plane compression to embed the secret information with *Key-2*, which can achieve high utilization ratio of redundant space with low distortion rate.

In the information extraction and secret image recovery phase, the receiver extracts the secret information and recovers the secret image based on the owned key.

3.1 Carrier Encryption

The camouflage encryption mainly performs the conversion of the plaintext image by performing coarse-grained sub-block matching and fine-grained data reversible conversion on the image carrier. The effect of camouflage encryption is mainly determined by the rationality of sub-block matching and the accuracy of conversion. At present, the camouflage encryption algorithm based on cluster matching achieves the high success rate of similar sub-block matching under low auxiliary information as much as possible and reduces the distortion by using translation and conversion. However, the algorithm has two problems. One is that it does not consider the edge distortion problem between sub-blocks, which may affect image quality. The other one is that it performs translation and truncation operations on sub-blocks with out-of-bounds pixels, which increases the pixel difference between sub-blocks, and results in image distortion. Moreover, the operations increase the volume of auxiliary information, which must be embedded into the camouflage image. In this paper, a new out-of-bounds pixel processing strategy is designed. This strategy can recover the pixel information only by recording whether the pixel is out of bounds, which optimizes the image edge distortion without adding auxiliary information. Fig. 2 shows the framework of the camouflage encryption algorithm proposed in this paper.

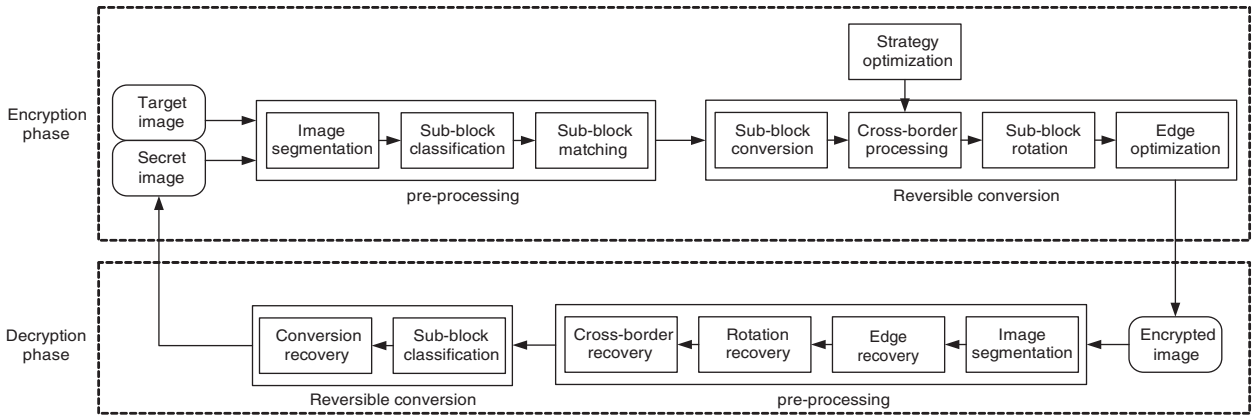

Figure 2: The framework of Camouflage encryption algorithm

Given the secret image S and the target image C, S will be transformed into an encrypted image which is similar to C as follows:

(1) Image segmentation. The secret image S and the target image C are respectively divided into N sub-blocks of the size $l \times w$ that do not overlap. Sub-blocks $B_i\,(1 \le i \le N)$ and $T_i\,(1 \le i \le N)$are obtained, the pixel values of which are represented as $B = \{p_1, p_2, \ldots, p_{l\times w}\}$ and $T = \{p'_1, p'_2, \ldots, p'_{l\times w}\}$.

(2) Sub-block classification. Sub-blocks $B_i\,(1 \le i \le N)$ and $T_i\,(1 \le i \le N)$ are clustered into K classes according to the standard deviation σ of the pixel values as a constraint with the method in [29].
(3) Sub-block matching. The secret image S and the target image C respectively sort the sub-blocks of each class in raster order, generate a composite index in the order of classes. The composite index is used as a mapping rule to complete the approximate matching between sub-block pairs. The corresponding sub-block pairs will have a similar standard deviation σ.
(4) Sub-block conversion. The sub-block of the image S is subjected to shift conversion, and the shift amount is an integer portion of the difference between the mean of the pixel values of the image S and the C's sub-block pairs.
(5) Cross-border processing. For out-of-bounds pixels, we design a new processing strategy in combination with the modulo operation as shown in Eq. (1):

$$p_{after\_i} = \begin{cases} 0 - p_i, & if\ \ p_i \le 0 \\ (255 - p_i)\bmod 256, & if\ \ p_i > 255 \\ p_i, & otherwise \end{cases}. \tag{1}$$

where p_i and p_{after_i} represent image pixels before and after conversion processing, respectively. Under this strategy, the processed overflow data is close to pixel 255 boundary, and the underflow data is close to pixel 0 boundary. Under the influence of as little fluctuation as possible, the out-of-bounds data is controlled within the range of 0–255 pixel, and the auxiliary information is not required to record the sign of the out-of-bounds, only need to mark whether it is out-of-bound in the position map.

In this paper, we set the corresponding position map MP, where $MP_i = 1$ indicates that the pixel point is the out-of-bound position, and $MP_i = 0$ indicates that the pixel point does not need to be processed. Obviously, the position map is a sparse binary matrix that can be effectively compressed using an entropy encoder to obtain Z_MP. According to the location map MP, the strategy can be reversible recovery by

$$p_i = \begin{cases} 0 - p_{after\_i}, & if\ \ mp_i == 1 \& p_{after\_i} < 255 - p_{after\_i} \\ 511 - p_{after\_i}, & if\ \ mp_i == 1 \& p_{after\_i} \ge 255 - p_{after\_i} \\ p_{after\_i}, & if\ \ mp_i == 0 \end{cases}. \tag{2}$$

(6) Sub-block rotation. Each sub-block is rotated in four directions of 0°, 90°, 180° and 270° in turn, and the direction with the smallest $RMSE$ between the target sub-blocks T is selected to be rotated. The final rotation direction V_dire of the sub-blocks is recorded for decryption operation. The rotation of the sub-block does not affect the statistical characteristics of the sub-block including the mean value and the standard deviation but can improve the matching degree of the data in the sub-block.
(7) Edge optimization. In order to improve the smoothness of the sub-block edge, the prediction algorithm and neighborhood data of edge pixels are used for edge optimization.

The following edge pixel $p(i,j)$ is taken as an example, and the mean value $pm(i,j)$ of the vertical neighborhood data of the edge pixel $p(i,j)$ can be calculated by

$$pm(i,j) = \left\lfloor \frac{p(i-1,j) + p(i+1,j)}{2} \right\rfloor. \tag{3}$$

Taking into account the pixel correlation within the sub-block, the prediction value $p_p(i,j)$ of $p(i,j)$ obtained by the prediction algorithm (such as the MED algorithm) and the corresponding prediction error $d(i,j) = p(i,j) - p_p(i,j)$ is further used, and the final replacement value$pr(i,j)$ is obtained can be calculated by

$$pr(i,j) = pm(i,j) + d(i,j). \tag{4}$$

From the above, we finally get the encrypted image $SC = E(S)$ that completely conceals the original plaintext information with the target plaintext information. At the same time, the auxiliary information, including the sub-block shift transformation, the out-of-boundary data-position map *MP*, the sub-block rotation direction, etc., are compressed and encrypted by a common encryption algorithms such as AES with an encryption key *Key-1*. And then the encrypted auxiliary information will be embedded into the encrypted image *SC* by RDH method, which is described in 3.2.

3.2 Information Embedding

Since the encrypted image *SC* is still a meaningful image, the existing reversible data hiding algorithms are applicable to this step. In general, it is not considered the influence of embedding on the distortion of encrypted image. However, considering that *SC* is in pseudo plaintext state, we still need a high embedding rate and low distortion for reversible data hiding algorithm. The modification of the LSB plane has the advantages of simplicity and little effect on the carrier. We introduce a reversible data hiding algorithm based on bit plane compression. Unless otherwise stated, the algorithm uniformly uses the raster scan order to read the matrix information. The specific embedding process is described as follows.

(1) Bit plane separation. Each plane can be separated and obtained by:

$$b_{sc}(k) = \left\lfloor \frac{p_{SC}}{p^k} \right\rfloor \bmod 2, \quad k \in \{7,6,\ldots,0\}. \tag{5}$$

where p_{SC} is the pixel value of the encrypted image, k is the corresponding layer number of the bit plane, b_{SC} is the binary corresponding to the *k-th* layer of the pixel value.

In the natural binary code, there are big differences in the codewords of some adjacent data, so the natural binary code can be converted to the Gray code by:

$$g_i = \begin{cases} b_i, & if\ i == 7 \\ b_i \oplus b_{i+1}, & if\ 0 \le i \le 6 \end{cases}. \tag{6}$$

Among them, the natural binary code is defined as $B = (b_7, b_6, \ldots, b_0)$, the corresponding Gray code is defined as $G = (g_7, g_6, \ldots, g_0)$, and $\oplus$ is defined as XOR operation.

(2) Sub-block tag and sub-block compression. The matrix is divided into a series of non-overlapping blocks of size $s_1 \times s_2$. According to the characteristics of uneven distribution of

elements, sub-blocks can be divided into five types, including all-zero sub-blocks, all-one sub-blocks, sparse sub-blocks with mostly zeros, sparse sub-blocks with mostly ones, and remainder common sub-blocks. The sub-blocks can be classified into types 1 to 5. To distinguish type 5 from other types, the information of type 5 is stored by the binary sequence *Mark_type*, where the 0 mark type 5, 1 is set to mark other types. The relevant information of the sub-block marks are in Tab. 1. e_0 represents the number of 0 elements in the sub-block, and e_1 represents the number of 1 elements in the sub-block, $e = \min\{e_0, e_1\}$. e_t is the threshold for the number of sparse elements and will be explained as follows.

For the sub-blocks whose *Mark_type* is marked as 1 in the sequence, we further implement the corresponding compression embedding method for different types of sub-blocks.

For the sub-blocks of type 1 and type 2, the original information can be completely recovered according to the marked block type information, and no additional auxiliary information is needed, that is, corresponding available embedded space $v_1 = s_1 \times s_2 - 2$.

Table 1: Block labeling rules

Block type	Type description	Type tag
1	$e_1 = 0$	00
2	$e_0 = 0$	01
3	$e_0 > e_1 \& 1 \le e_1 \le e_t$	10
4	$e_1 > e_0 \& 1 \le e_0 \le e_t$	11
5	$e > e_t$	–

The sub-blocks of type 3 and type 4 are binary sparse matrices. In order to completely recover the relevant information in the extraction phase. In addition to the block type information, information of the sparse elements also need to be recorded. Since the sub-block values are only 0 and 1, and the sparse element values can be determined from the type information, only the number of sparse elements and the relative positions in the sub-blocks need to be recorded. Set the minimum remaining space $v_{2\min} = s_1 \times s_2 - 2 - e\left\lceil \log_2 (s_1 \times s_2) \right\rceil - \left\lceil \log_2 e \right\rceil$as the type indicator. Where $v_{2\min} > 0$, the type belongs to type 3 or type 4; otherwise, it is classified as type 5. From the above, the corresponding sparse element data threshold can be defined as follows:

$$e_t = argmax\,(v_{2\min} \ge 0)\,. \tag{7}$$

(3) Information embedding. First, each sub-block information is read in the raster scanning order to obtain its embedded space v_i and M_i of size v_{mi} after *Mark_type* lossless encoding (such as run length encoding);

Secondly, the type of information tag and the secret information is operated in the first four types of sub-blocks in order, and the sub-blocks are obtained after being encrypted, and the type 5 sub-block is not operated. For the lowest bit plane, if the embedded space satisfies $v_1 \ge L$ (L is the length of embedded information, which includes encrypted auxiliary information and secret information), the embedding is completed. Otherwise, the remaining $L - v_1$ secret information and M_1 are transmitted to the adjacent bit plane;

Finally, if the bit plane embedded space satisfies $v_i \ge L + v_{m(i-1)}$, the information embedded is performed according to step two; if $v_i < L + v_{m(i-1)}$ is satisfied, the secret information of size $L_i = v_i - v_{m(i-1)}$ is embedded in the bit plane, and the remaining space is used for embedding M_{i-1}, and

the information of the remaining length $L - L_i$ is embedded by the adjacent bit plane, repeatedly until the embedding is completed, and finally the carrier plane is obtained and combined to form the encrypted carrier SC_{em}.

In this process, the length of embedded information L_i and $v_{m(i-1)}$ of each bit plane and the highest embedded bit plane M_i are saved as the embedded key *key_2*, and $v_{m0} = 0$; when the secret information cannot be completely hidden by the ordered integer sub-blocks, the remaining space of the last sub-block can be filled with random bit data.

3.3 Information Extraction and Secret Image Recovery

When the receiver obtains the encrypted carrier SC_{em}, the information can be extracted without distortion and the carrier can be reversibly restored according to the key *key_1 and key_2*. Information extraction and secret image recovery are the inverse process of carrier encryption and information embedding.

According to the embedded key *key*_2, the secret parameters used in the embedding process are obtained, including the embedding information L_i and $v_{m(i-1)}$. of each bit plane and the M_i of the highest embedding bit plane.

(1) Bit plane separation. After Gray coding and separation, each bit plane data $g_{sc}(k)$ corresponding to the encrypted carrier is obtained. According to the obtained hidden parameters, the layer-by-layer extraction is performed from the higher bit plane to the lower bit plane.
(2) Mark identification. According to the *Mark_type* obtained by M_i decompression, type 5 and other types can be distinguished according to Tab. 1.
(3) Information extraction. For different types of sub-blocks, according to the compression embedding method used, the corresponding data extraction and sub-block recovery methods are as follows:
    (a) For the sub-blocks of type 1 and type 2, except for the first two marked bits, the other bits are encrypted information. Relevant information can be extracted by reading them in order of embedding sequence. And the sub-blocks are restored to all 0 or all 1 according to the sub-block type;
    (b) For the sub-blocks of type 3 and type 4, the number of sparse elements and the encoded position information are obtained according to the agreed embedding position, the rare element positions are obtained by de-encoding the position information, and the encrypted information is obtained by reading the rest of the positions based on the embedding order. The atomic blocks are recovered with the type information;
    (c) The first L_i data extracted in sequence is the embedded encrypted information and the next data $v_{m(i-1)}$ are M_{i-1} data of the adjacent low plane. Data $Mark\_type_{i-1}$ can be recovered after decompression. The encrypted information can be extracted and stitched layer by layer, and the bit plane can be recovered.
(4) Bit plane recovery. The bit plane recovered from above is obtained by Gray coding, and the corresponding decoding can be used to recover the original bit plane without loss. Finally, the secret information and the encrypted image are obtained.
(5) Secret image recovery. According to the *key_1*, the auxiliary information is decrypted and decompressed, including sub-block shift transformation, out-of-boundary data location map, sub-blocks rotation direction. The secret image recovery methods are as follows:

(a) Edge recovery. For the edge modification operation in the encryption step, $p(i,j)$can be recovered by formula (8) according to $pr(i,j)$.

$$p(i,j) = p_p(i,j) + pr(i,j) - pm(i,j) \tag{8}$$

Among them, because the neighborhood data remains unchanged, the mean value $pm(i,j)$ and the predicted value $p_p(i,j)$ can be directly calculated without auxiliary information. If the neighborhood data is also replaced by pixels, it can be recovered in the reverse order of replacement.

(b) Rotation recovery. Based on the related information of sub-block rotation direction obtained from the auxiliary information, each sub-block is rotated in the opposite direction to realize rotation recovery.

(c) Out-of-bounds data recovery. According to the recovered location map MP, the original out-of-bounds data position can be identified. With the ciphertext data and location information, and the original cross-border data information can be directly recovered according to formula (2).

(d) Conversion recovery. The reversible transformation of a sub-block is independent of its class, and the sub-block's shift transformation in the auxiliary information can be directly used to restore.

4 Experimental Results and Analysis

4.1 Carrier Encryption Experiment

We use the image in Fig. 3a is the secret carrier and the image in Fig. 3b as the corresponding target carrier. In order to select the appropriate block size, we use the block size as a variable to carry out the experiment. The corresponding experimental results are shown in Figs. 3c–3h and Tab. 2. From a subjective point of view, when the block size gradually becomes larger, the quality of the camouflage image gradually decreases; from the objective data provided in Tab. 2, the block size is negatively correlated with the quality of the camouflaged image and is positively correlated with the number of auxiliary information. Auxiliary information as decryption related information usually can be embedded in camouflage image by reversible steganography algorithm or be stored directly after encryption. In order to balance the quality of the image and the amount of auxiliary information, in the following experiments, we set the block size to 4×4.

In order to further analyze the influence of the choice of the target object on the encryption algorithm, we take Fig. 4a as the secret image and Figs. 4b–4f as the target image for experiments. The camouflaged encrypted images obtained by the experiment are shown in Figs. 4g–4k, and the corresponding experimental performance data is shown in Tab. 3. The experimental results show that the secret image has a good effect on camouflage encryption of most target image, and has good versatility. Since the converted sub-block pair is approximately matched based on the sub-block variance as the feature, the reversible conversion is based on the translation of the target sub-block mean. When the image texture is similar, the encrypted sub-blocks have similar variances and mean values corresponding to the target sub-blocks. So it can achieve higher SSIM and better similarity. However, it also leads to the relatively poor conversion effect when the texture complexity gap between the two images is large, that is, when the variance gap of the matching sub-blocks is large. For example, the SSIM of encryption results with Baboon as the target image is not good. Therefore, although the encryption algorithm used in this paper has

good versatility, it can achieve a better camouflage encryption effect when the texture complexity of the secret image and target image is similar.

Figure 3: (a) Secret image (b) target image (c) Camouflage image with a block size of 2×2 (d) Camouflage image with a block size of 3×3 (e) Camouflage image with a block size of 4×4 (f) Camouflage image with a block size of 5×5 (g) Camouflage image with a block size of 6×6 (h) Camouflage image with a block size of 8×8 mouflage image with a block size

Table 2: Encryption effects under blocks of different sizes

Block size	RMSE	SSIM	Auxiliary information(bpp)
Fig. 3c 2×2	5.74	0.9905	2.1992
Fig. 3d 3×3	7.77	0.9826	1.0119
Fig. 3e 4×4	10.39	0.9690	0.5715
Fig. 3f 5×5	11.88	0.9598	0.3628
Fig. 3g 6×6	13.56	0.9464	0.2573
Fig. 3h 8×8	16.68	0.9227	0.1426

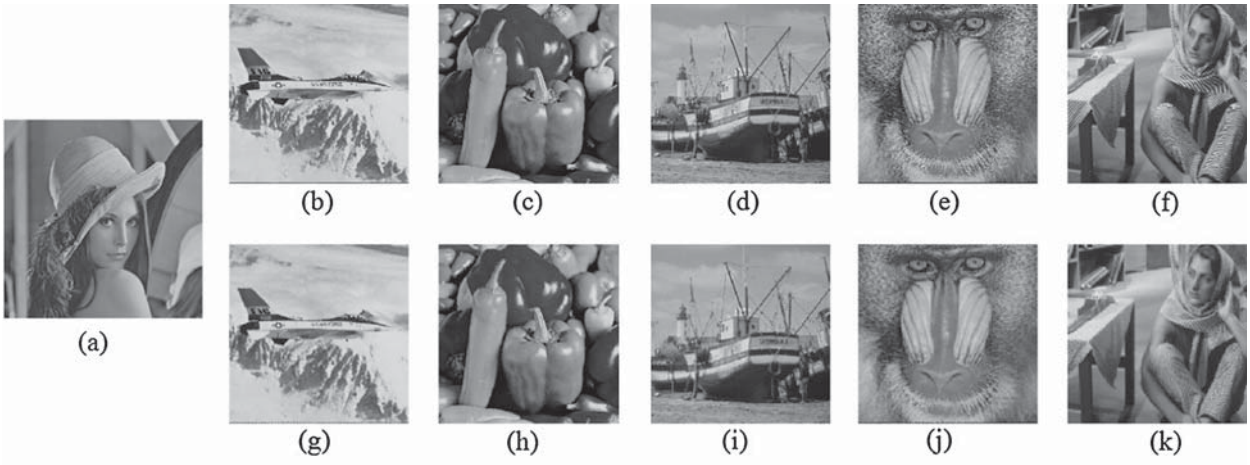

Figure 4: Experimental image and encrypted image (a) secret image lena (b) target image airplane (c) target image peppers (d) target image baboon(e) target image boats (f) target image barbara (g) encrypted image airplane (h) encrypted image peppers (i) encrypted image baboon (j) encrypted image boats (k) encrypted image barbara

It can be seen from Fig. 4 that the camouflage encrypted images are similar with the corresponding target images and have high SSIM between them in Tab. 3, which means that the original secret image content is totally by the camouflage encrypted image content. Even if the attacker recognizes the camouflage, without the encryption key *Key-1* of AES, it is also unable to decrypt the auxiliary information that is necessary for recovering the original secret image. And the attacker cannot get any information of the secret image as shown in Fig. 5.

Table 3: Encryption effect under different target image

Target -camouflage image	RMSE	SSIM	Auxiliary information (bpp)
Figs. 4b–4g	10.81	0.9707	0.5713
Figs. 4c–4h	9.35	0.9866	0.5949
Figs. 4d–4i	10.95	0.9778	0.5840
Figs. 4e–4j	21.27	0.8575	0.5751
Figs. 4f–4k	16.86	0.9507	0.5925

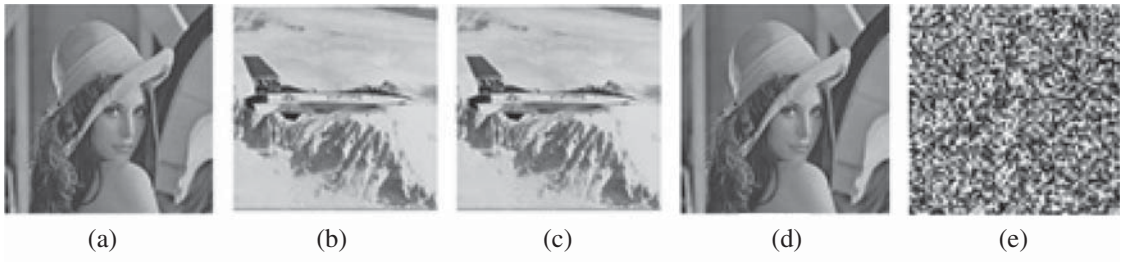

Figure 5: Experimental results of encrypted image (a) secret image lena (b) target image airplane (c) encrypted image airplane (d) decrypted image (with *Key-1*) (e) decrypted image (with wrong key)

We compare the encryption algorithm with current similar algorithms [27,29] to further analyze the performance of the encryption algorithm. Since the existing algorithms usually use color images as the encryption object, we use Fig. 6a as the secret image and Fig. 6b as the target image. We decompose the three channels (R, G, B) of the color image into three gray-scale images for camouflage encryption and then recombine for experiments. Among them, according to the algorithm described in [27], we use the block size 8×8 with the best effect and 4×4 block size for both [29] and this paper.

The experimental results are shown in Figs. 6c–6i and Tab. 4. It can be seen from the partial enlarged images in Figs. 6f–6i that the scheme used in this paper can significantly reduce the jigsaw effect of the image and reduce the distortion from the edge to the whole image. The data in Tab. 4 is also well verified the result, and the additional data of images are also significantly reduced. It can be seen that the encryption effect achieved by the algorithm in this paper is better than the algorithms in [27,29]. The edge optimization and out-of-bounds pixel processing strategies used in the encryption algorithm have a good effect on the camouflage encryption effect. However, considering that the encryption algorithm in this paper does not consider the color channel correlation for color images, the encryption effect has a lot of room for improvement.

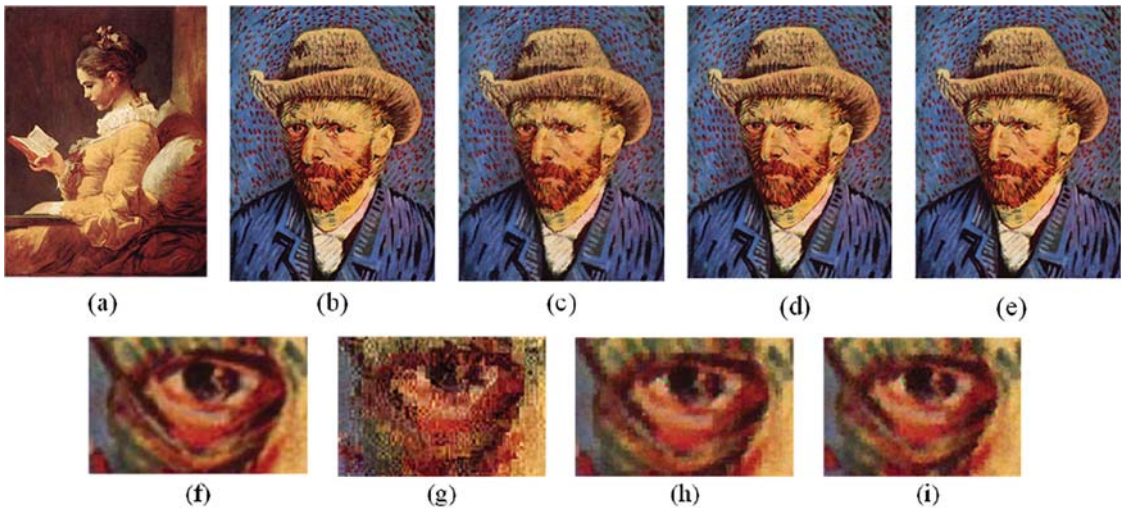

Figure 6: (a) Secret image (b) target image (c) Camouflage image generated by algorithm [27] (d) Camouflage image generated by algorithm [29] (e) Camouflage image generated by this paper (f)Part of the target image (g) part of camouflage image generated by algorithm [27] (h) part of camouflage image generated by algorithm [29] (i) part of camouflage image generated by this paper

Table 4: Encryption effect of camouflage image and target image

Algorithm	RMSE	SSIM
Reference [27] Fig. 5c	22.38	0.3562
Reference [29] Fig. 5d	13.85	0.6535
This paper Fig. 5e	11.07	0.7792

4.2 Data Hiding Experiment

In the experiment, the compression embedding of data is carried out from the low-bit plane to the high-bit plane to reduce the effect of embedding on the whole image. In order to make full use of the redundant space of camouflage image, we also block the carrier in the data hiding stage according to the size of 4×4. It can be calculated by Eq. (7), and $e_t = 3$.

We use Fig. 7(a) (the camouflage encrypted image in Fig. 3e) as a steganographic carrier, and the 64×64 size Fig. 7b as the secret information to carry out information embedding and extraction experiments. The corresponding results are shown in Fig. 7. It can be seen from the experimental results that both the encrypted carrier image and the extracted secret information have high quality. The PSNR of the encrypted carrier image is 71.14 dB for the camouflage encrypted image, and the extracted data image is relative to the embedded data image, $PSNR = \inf$. So the steganographic algorithm proposed in this paper can recover embedded information without distortion and is completely reversible.

In order to better explain the performance of the steganographic algorithm, this section starts from the embedding capacity and imperceptibility of steganography and conducts performance

testing and comparative experiments on encrypted carrier images. The experimental carrier is the corresponding image of Fig. 4 obtained in the encryption phase.

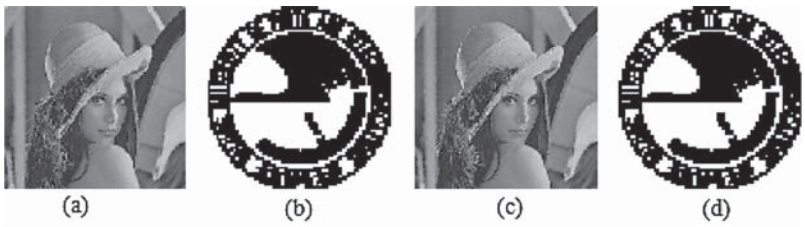

Figure 7: (a) Camouflage encrypted image (b) embedded data (c) encrypted carrier (PSNR = 71.14 dB) (d) extracted data (PSNR = Inf)

First, we use the generated camouflage encrypted images in Figs. 4g–4k as the carrier object, and separate the bit planes under the Gray coding and the ordinary binary coding used in this paper. Then we perform independent embedding steganography on the low 4-bit planes to quantify the available space of the steganography algorithm under different encodings. The embedding capacity of each layer of the camouflage encrypted image is shown in Tab. 5. Obviously, under the same-level bit plane, the available bit plane redundancy space for Gray coding is greatly improved, which is similar to the embedding capacity of the adjacent high-level bit plane under ordinary binary.

Table 5: Bit plane steganographic capacity under different coding

Camouflage encrypted image	Lowest plane		Second low plane	
	This paper	Ordinary binary	This paper	Ordinary binary
Airplane	2365	766	16437	2220
Peppers	2300	697	16342	2496
Baboon	2297	796	12743	2181
Boat	2484	772	17575	2333
Barbara	2176	793	15138	2330
Camouflage encrypted image	Third low plane		forth low plane	
	This paper	Ordinary binary	This paper	Ordinary binary
Airplane	54769	17012	100219	56437
Peppers	52942	15804	98917	55640
Baboon	41079	12587	82415	42270
Boat	53678	16910	102095	55634
Barbara	49969	14453	96055	50360

In order to better evaluate the performance of the algorithm, we compare the proposed scheme with other similar schemes. Considering the particularity of the algorithm in this paper, we choose the existing classical reversible steganography algorithm [23,24] to apply to the camouflage encrypted images generated in this paper, in order to make a more reasonable comparison of experiments. Tab. 6 shows the comparison data between the algorithm and the [23,24]. Compared

with the algorithm in [23,24], the algorithm proposed in this paper can achieve better image quality and superior performance under the same embedding rate.

Table 6: Performance comparison between this paper and the reference

Camouflage encrypted image	Capacity (bit)	This paper PSNR (dB)	Reference [23] PSNR (dB)	Reference [24] PSNR (dB)
Airplane	20000	58.87	58.03	58.69
Peppers	20000	58.94	57.99	57.86
Baboon	20000	56.64	57.92	56.53
Boat	20000	59.35	58.14	58.02
Barbara	20000	58.53	58.05	57.91
Average	20000	58.47	58.03	57.80

5 Conclusion

In this paper, a reversible steganography algorithm based on image camouflage encryption and bit plane compression is proposed, and an edge optimization method based on the prediction algorithm is designed. The experimental results show that the method can achieve a good balance between embedding capacity and image quality, effectively improve the encryption quality, and have good performance.

Funding Statement: This work was supported in part by the National Key R&D Program of China (2019YFB1406504), and the National Natural Science Foundation of China (U1836108, U1936216, 62002197).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Q. Mo, H. Yao, F. Cao, Z. Chang and C. Qin, "Reversible data hiding in encrypted image based on block classification permutation," *Computers Materials & Continua*, vol. 59, no. 1, pp. 119–133, 2019.

[2] J. Wang, L. Huang, Y. Zhang, Y. Zhu, J. Ni *et al.,* "An effective steganalysis algorithm for histogram-shifting based reversible data hiding," *Computers, Materials & Continua*, vol. 64, no. 1, pp. 325–344, 2020.

[3] K. Ma, W. M. Zhang, X. F. Zhao, N. H. Yu and F. H. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.

[4] S. Yi and Y. C. Zhou, "Binary-block embedding for reversible data hiding in encrypted images," *Signal Processing*, vol. 133, no. 10, pp. 40–51, 2017.

[5] W. Zhang, K. Ma and N. Yu, "Reversibility improved data hiding in encrypted images," *Signal Processing*, vol. 94, no. 6, pp. 118–127, 2014.

[6] C. W. Shiu, Y. C. Chen and W. Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion," *Signal Processing: Image Communication*, vol. 39, no. 9, pp. 226–233, 2015.

[7] X. C. Cao, L. Du, X. X. Wei, D. Meng and X. J. Guo, "High capacity reversible data hiding in encrypted images by patch-level sparse representation," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1, 2015.

[8] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

[9] W. Hong, T. S. Chen and H. Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.

[10] J. Yu, G. Zhu, X. Li and J. Yang, "An improved algorithm for reversible data hiding in encrypted image," in *Proc. IWDW*, Shanghai, SH, China, pp. 384–394, 2012.

[11] W. Hong, T. S. Chen, J. Chen, Y. H. Kao, H. Y. Wu *et al.,* "Reversible data embedment for encrypted cartoon images using unbalanced bit flipping," in *Proc. ICSI*, Harbin, HB, China, pp. 208–214, 2013.

[12] X. Liao and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *Journal of Visual Communication & Image Representation*, vol. 18, no. 12, pp. 21–27, 2015.

[13] Z. L. Yang, X. Q. Guo, Z. M. Chen, Y. F. Huang and Y. J. Zhang, "RNN-stega: Linguistic steganography based on recurrent neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1280–1295, 2018.

[14] Z. L. Yang, S. Y. Zhang, Y. T. Hu, Z. W. Hu and Y. F. Huang, "VAE-Stega: Linguistic steganography based on variational auto-encoder," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 880–895, 2020.

[15] Z. L. Liu and C. M. Pun, "Reversible data-hiding in encrypted images by redundant space transfer," *Information Sciences*, vol. 434, no. 12, pp. 188–203, 2018.

[16] M. S. A. Karim and K. Wong, "Universal data embedding in encrypted domain," *Signal Processing*, vol. 94, no. 6, pp. 174–182, 2014.

[17] X. Wu, J. Weng and W. Q. Yan, "Adopting secret sharing for reversible data hiding in encrypted images," *Signal Processing*, vol. 143, no. 9, pp. 269–281, 2018.

[18] M. Q. Zhang, Y. Ke and T. T. Su, "Reversible steganography in encrypted domain based on LWE," *Journal of Electronics & Information Technology*, vol. 38, no. 2, pp. 354–360, 2016.

[19] S. J. Xiang, X. R. Luo and S. X. Shi, "A novel reversible image watermarking algorithm in homomorphic encrypted domain," *Chinese Journal of Computers*, vol. 39, no. 3, pp. 571–581, 2016.

[20] X. Zhang, J. Long, Z. Wang and H. Cheng, "Lossless and reversible data hiding in encrypted images with public-key cryptography," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 8, no. 6, pp. 1622–1631, 2016.

[21] M. Li, D. Xiao, Y. Zhang and H. Nan, "Reversible data hiding in encrypted images using cross division and additive homomorphism," *Image Communication*, vol. 39, no. 10, pp. 234–248, 2015.

[22] D. Xiao, Y. Xiang, H. Zheng and Y. Wang, "Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism," *Journal of Visual Communication and Image Representation*, vol. 45, no. 2, pp. 1–10, 2017.

[23] A. V. Subramanyam, S. Emmanuel and M. S. Kankanhalli, "Robust watermarking of compressed and encrypted JPEG2000 images," *International Journal of Computer Trends & Technology*, vol. 14, no. 3, pp. 703–716, 2013.

[24] F. Huang, J. Huang and Y. Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2777–2789, 2016.

[25] S. Yi, Y. Zhou and Z. Hua, "Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion," *Signal Processing: Image Communication*, vol. 64, no. 3, pp. 78–88, 2018.

[26] I. Lai and W. H. Tsai, "Secret-fragment-visible mosaic image—A new computer art and its application to information hiding," *IEEE Transactions on Information Forensics & Security*, vol. 6, no. 3, pp. 936–945, 2011.

[27] Y. L. Lee and W. H. Tsai, "A new secure image transmission technique via secret-fragment-visible mosaic images by nearly reversible color transformations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 695–703, 2014.

[28] W. Zhang, H. Wang, D. Hou and N. Yu, "Reversible data hiding in encrypted images by reversible image transformation," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1, 2016.

[29] D. Hou, W. Zhang and N. Yu, "Image camouflage by reversible image transformation," *Journal of Visual Communication and Image Representation*, vol. 40, no. 6, pp. 225–236, 2016.

[30] D. Hou, C. Qin, N. Yu and W. Zhang, "Reversible visual transformation via exploring the correlations within color images," *Journal of Visual Communication & Image Representation*, vol. 53, no. 11, pp. 134–145, 2018.