

# Reversible Visual Transformation via Exploring the Correlations within Color Images

Dongdong Hou, Chuan Qin, Nenghai Yu, Weiming Zhang\*

*School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China.*

---

## Abstract

Reversible visual transformation reversibly transforms a secret image to a freely-selected target image and gets a camouflage image similar to the target image, which has been proved to be very useful in such two applications: privacy protection of images and reversible data hiding in encrypted images. Now, a new reversible transformation technique for color images is proposed by exploring and utilizing the correlations among three color channels and inside each color channel. The amount of the accessorial information for recording transformation parameters is largely reduced. Therefore, the visual quality of the created camouflage image is much improved by dividing the secret image and the target image into even smaller blocks for transformation.

*Keywords:* reversible visual transformation, image camouflage, image encryption, reversible data hiding.

---

## 1. Introduction

Nowadays, images are frequently stored and transmitted on the Internet. However, some images may contain private or confidential information, which should be protected from leakages. Therefore, many methods have been proposed for the purpose of secure image transmission and secure image storage, among which two common approaches are encryption and data hiding.

Although encryption solves the privacy problem in a certain extent, the messy codes of ciphertext with special form are easy to cause the attention of attackers who plan to breakout the accounts of encryption users. However, data hiding technology embeds message into covers such as image, audio or video, which not only protects the content of secret file, but also hides the communication process itself to avoid the attacker's attention.

Traditional highly secure data hiding methods [1–3] are suitable for embedding a small message into a large cover, e.g., an image. But in the applications of image transmission or storage, the image itself is

---

\*Corresponding author. Tel.: +86 0551 3600683

*Email addresses:* [houdong@mail.ustc.edu.cn](mailto:houdong@mail.ustc.edu.cn) (Dongdong Hou), [qc94@mail.ustc.edu.cn](mailto:qc94@mail.ustc.edu.cn) (Chuan Qin), [ynh@ustc.edu.cn](mailto:ynh@ustc.edu.cn) (Nenghai Yu), [zhangwm@ustc.edu.cn](mailto:zhangwm@ustc.edu.cn) (Weiming Zhang)

<sup>1</sup>This work was supported in part by the Natural Science Foundation of China under 61572452, Grant U1636201.

just the message. Therefore, large capacity data hiding technique is applied to hide images. Although large capacity hiding technique is hard to resist the detection of strong steganalysis [4–6], it will be very useful in the applications of protections of images.

Visual transformation can be viewed as a data hiding method having super large payloads, which embeds one image into another one with the same size. We transform the secret image to the target image, getting the camouflage image similar to the target image. From the camouflage image, the recipient can reconstruct the secret image. Note that, the existed visual transformation schemes [7–9] as well as the presented scheme is for images with lossless formats, among which PNG is the most popular lossless format adopted by many camera equipments such as iPhone and iPad.

Visual transformation technique sprouts from Lai *et al.*'s work [7]. But by Lai *et al.*'s method, the target image cannot be selected arbitrarily, which must be similar to the secret image. What is more, the visual quality of camouflage image is relatively poor. Then Lee *et al.* [8] adopt color transformation [10] to transform a secret image to a freely-selected target image, and greatly improve the visual quality of camouflage image. However, one of the main drawbacks of Lee *et al.*'s method is that the secret image cannot be losslessly reconstructed due to that the adopted color transformation [10] is not reversible.

To overcome the drawbacks of Lai *et al.*'s method and Lee *et al.*'s method, we present a reversible visual transformation (RVT) scheme in [9] to transform a secret image to a freely-selected target image by adopting reversible shift transformation. Before shift transformation, a non-uniform clustering algorithm is utilized to match secret blocks and target blocks, which largely reduces the amount of accessorial information (AAI) for recording indexes of secret blocks. With the method [9], not only is the visual quality of camouflage image improved a lot, but also the secret image can be restored losslessly.

In addition to secure image transmission and storage, the RVT scheme proposed in [9] can be also used in some other applications for privacy protection. For example, there comes a novel framework [11] for reversible data hiding in encrypted image (RDH-EI) by replacing encryption with the RVT scheme [9]. Note that RDH-EI is a protocol to protect the privacy of outsourced images in clouds by encryption and enables the cloud server to reversibly embed watermark in the encrypted image. Different from all the previous encryption based frameworks [12–14], in which the cloud server and the clients must agree on how to encrypt or decrypt their data, RVT-based framework enables the cloud server to embed data into the “encrypted image” easily by any reversible data hiding (RDH) methods for plaintext images. Thus a client-free scheme for RDH-EI can be realized, that is, the data embedding process executed by the cloud server is irrelevant with the processes of both encryption and decryption.

Although the method [9] improves the methods [7, 8] a lot, it does transformation for channels R, G and B of color images separately without exploring and utilizing the correlations among three color channels and inside each color channel, otherwise RVT scheme can be still improved greatly. In this paper, we firstly present an improved reversible shift transformation via assigning a mean class index for each block

according to its mean, and the correlations among these mean class indexes are high. Then we explore the correlations within transformation parameters to compress them. Thus AAI for recovering secret image is largely reduced, by which we improve the visual quality of camouflage image through dividing the secret image and the target image into even smaller blocks and transforming secret blocks to corresponding target blocks. Experimental results show that the presented method outperforms the previous methods [7–9] a lot.

The rest of this paper is organized as follows: Section 2 introduces the related work. The proposed method is elaborated in Section 3. Experimental results are shown in Section 4, and the paper is concluded with a discussion in Section 5.

## 2. Related arts

In the scheme of visual transformation, we divide secret image and target image into small blocks and transform secret blocks to corresponding target blocks to get a transformed image firstly, and then embed the accessorial information for recovering secret image into transformed image with RDH techniques such as [15–18] to yield ultimate camouflage image. At the receiver side, the receiver firstly restore transformed image from camouflage image after extracting these accessorial information, and then restore secret image from transformed image with the help of extracted accessorial information.

The presented RVT scheme is based on the RVT scheme proposed in [9], which consists of two main parts: reversible shift transformation and blocks matching, and is described as follows.

### 2.1. Reversible shift transformation

We divide secret image and target image into  $N$  non-overlapping blocks with the same size. Let the secret block  $\mathbf{P}$  be a set of pixels such that  $\mathbf{P} = \{p_1, p_2, \dots, p_n\}$  with symbol  $u$  representing its mean, and the target block  $\mathbf{P}' = \{p'_1, p'_2, \dots, p'_n\}$  with symbol  $u'$  representing its mean. The purpose of visual transformation technique is to make the mean and the standard deviation (SD) of each secret block be similar with those of the corresponding target block, by which the secret image will be masqueraded as the target image visually.

A shift transformation is presented in [9] by shifting each pixel of secret block with the amplitude  $(u' - u)$ . To keep the transformation reversible, the amplitude  $(u' - u)$  is rounded to be the closest integer, namely

$$\Delta u = \text{round}(u' - u). \quad (1)$$

$\Delta u$  is recorded for shifting transformed block back to secret block. To reduce AAI for recording  $\Delta u$ , we quantize it by

$$\Delta u' = \begin{cases} 8 \times \text{round}(\frac{\Delta u}{8}), & \text{if } \Delta u > 0 \\ 8 \times \text{floor}(\frac{\Delta u}{8}) + 4, & \text{if } \Delta u < 0 \end{cases}, \quad (2)$$

where function  $\text{floor}(x)$  means rounding  $x$  to be the nearest integer not larger than it. With the quantized  $\Delta u'$ , the transformed block  $\mathbf{P}'' = \{p''_1, p''_2, \dots, p''_n\}$  is generated as

$$p''_i = p_i + \Delta u'. \quad (3)$$

Since  $\Delta u'$  is quantified by the step length 8 with Eq. (2),  $\Delta u''$  is recorded as the parameter for restoring secret block, where

$$\Delta u'' = |\Delta u'|/4, \quad (4)$$

which is in the range of 0-64. Because the correlations among  $\Delta u''$ 's are weak, it will cost much accessorial information to record  $\Delta u''$ 's.

Obviously  $\mathbf{P}''$  will have a similar mean with  $\mathbf{P}'$  by the shift transformation Eq. (3). To make  $\mathbf{P}''$  be more similar to  $\mathbf{P}'$  we further rotate it into one of the four directions  $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ . The optimal rotated direction is the one which makes the minimum root mean square error (RMSE) between the rotated version and the corresponding target block.

## 2.2. Blocks matching

The transformation Eq. (3) shifts the mean of secret block to the mean of target block, but each secret block should be paired with a proper target block before shift transformation, that is pairing each secret block with a target block having a similar SD. In order to reduce AAI for recording the mapping among secret blocks and target blocks, a non-uniform clustering algorithm is proposed for matching secret blocks and target blocks according to their SDs, which is described as follows.

Firstly, all the SDs of secret blocks (and thus the corresponding blocks) are clustered into  $K$  classes (called SD class) by  $K$ -means to ensure that the SDs in the  $i$ th SD class are smaller than those in the  $j$ th SD class when  $1 \leq i < j \leq K$ . According to secret SD classes' volumes and the scanning order, we cluster target blocks into  $K$  classes and set each target SD class to include the same volume as the corresponding secret SD class. If the  $i$ th secret SD class includes  $n_i$  blocks, then the first  $n_1$  target blocks with the smallest SDs are clustered into the first SD class and the next  $n_2$  target blocks with the smallest SDs of the rest of target blocks are clustered into the second SD class, and so on, until all target blocks are clustered. Therefore, both secret blocks and target blocks are clustered into  $K$  classes according to their SDs.

To pair secret blocks and target blocks, each block is assigned with a compound index. That is we scan blocks in the order and the  $j$ th block belonging to the  $i$ th SD class is labeled as the  $i_j$  ( $1 \leq i \leq K$ ,  $1 \leq j \leq n_i$ ). With assigned compound indexes, the one to one map among secret blocks and target blocks is established. The  $i_j$ th secret block is transformed to the  $i_j$ th target block and then the  $i_j$ th target block will be replaced by the corresponding transformed block, by which a transformed image will be generated.

Shift and rotation of secret block will not change its SD, thus SDs of transformed blocks will be the same as those of corresponding secret blocks. At the receiver's side, transformed blocks can be clustered into the

same  $K$  classes as secret blocks, and assigned compound indexes equaling to those of the target blocks at the same position. Secret compound indexes need to be recorded for recovery, to do that we only need record secret SD class indexes in the scanning order. These SD class indexes can be compressed efficiently due to that the distribution of SDs from the divided small blocks is concentrated.

### 3. Proposed method

As mentioned before, secret image and target image are divided into small blocks for transformation. With the smaller block size, we can get a transformed image with better quality. However, the smaller the block size is, the more accessorial information should be embedded into the transformed image which will introduce two problems. One is that the RDH scheme cannot reach an enough capacity to accommodate these accessorial information. The other is that the more information is embedded into the transformed image, the worse quality the ultimate camouflage image will result in. Therefore, as long as AAI for recovering secret image is reduced, we can improve the quality of camouflage image by setting the smaller block size or embedding the less information into transformed image.

The method [9] improves Lee *et al.*'s method [8] a lot by setting the smaller block size, mainly thanks to adopting non-uniform clustering algorithm for matching secret blocks and target blocks, by which AAI for recovering each secret block is reduced. However, the method [9] as well as Lee *et al.*'s method [8] does transformation for three channels of color images respectively without considering the correlations among three color channels and inside each channel, so it will cost much accessorial information to record their transformation parameters. But the correlations exist obviously within color images. By the method [9], the transformation parameters mainly include the optimal rotated directions, the secret SD class indexes and the shift amplitudes  $\Delta u$ 's in Eq. (4). In this section, based on the work [9] we will explore and utilize the correlations within color images to compress the optimal directions and the secret SD class indexes. As for the shift amplitudes  $\Delta u$ 's in Eq. (4), an improved shift transformation is presented, by which instead of recording  $\Delta u$ 's we record new parameters (i.e., the mean class indexes) for recovery among which the correlations are higher. Therefore, AAI for recording the transformation parameters is greatly reduced by the presented method.

#### 3.1. Improved reversible shift transformation

Now, we give an improved shift transformation producing the less accessorial information, and at the same time keeping the correlations among the transformation parameters for compression. Different from Eq. (1), in the presented paper  $\Delta u$  is defined as the absolute difference from the mean of secret block to the mean of corresponding target block. That is

$$\Delta u = \text{round}(|u' - u|), \quad (5)$$

and then  $\Delta u$  is quantified by the step length 8 with

$$\Delta u' = 8 \times \text{round}\left(\frac{\Delta u}{8}\right). \quad (6)$$

With the rounded and quantified  $\Delta u'$ , each transformed block  $\mathbf{P}'' = \{p_1'', p_2'', \dots, p_n''\}$  is got by

$$p_i'' = \begin{cases} p_i + \Delta u', & \text{if } u' \geq u \\ p_i - \Delta u', & \text{if } u' < u \end{cases}. \quad (7)$$

Obviously, transformation Eq. (7) and transformation Eq. (3) can achieve the same result.

To shift transformed block back to secret block, we assign a mean class index for each block according to its mean. The mean class index of secret block (denoted by  $MS$ ) is calculated as Eq. (8) by quantifying its mean with the same step length 8 as that utilized in Eq. (6).

$$MS = \text{round}\left(\frac{u}{8}\right). \quad (8)$$

In the same way we assign a mean class index for transformed block denoted by  $MT$ .

Since  $\Delta u'$ ,  $MS$  and  $MT$  are quantified by the same step length 8 with Eq. (6) and Eq. (8) respectively,  $u$  being shifted to  $u''$  (the mean of transformed block) with the amplitude  $\Delta u$  means  $MS$  being shifted to  $MT$  with the amplitude  $\frac{\Delta u}{8}$ . That is to say

$$\Delta u' = 8 \times |MT - MS|. \quad (9)$$

At the receiver side, each secret block  $\mathbf{P} = \{p_1, p_2, \dots, p_n\}$  can be restored by

$$p_i = \begin{cases} p_i'' - \Delta u', & \text{if } MT \geq MS \\ p_i'' + \Delta u', & \text{if } MT < MS \end{cases}. \quad (10)$$

Because  $\Delta u'$  can be restored from  $MS$  and  $MT$  with Eq. (9), and  $MT$  can be recalculated from transformed block with Eq. (8), we only need to record  $MS$  as the parameter instead of  $\Delta u'$  for shifting transformed block back to secret block. Obviously,  $MS$  is in the range of 0-32, which is half of the range of  $\Delta u''$  utilized in Eq. (4). More importantly, the correlations among  $MS$ s are much higher than the correlations among  $\Delta u''$ s. Indeed,  $MS$ s from R, G and B channels of a color secret image compose a special image that is sampled down from the secret image and with the sampled pixels scaled to 0-32. Therefore,  $MS$ s are much easier to be compressed than  $\Delta u''$ s, and we will discuss how to compress  $MS$ s later.

In the improved method, in addition to the SD class index utilized in the work [9], we assign another index, i.e., the mean class index for each block. The mean class indexes are utilized for shifting transformed blocks back to secret blocks, and the SD class indexes are utilized for blocks matching. Now, we give a simple example of blocks matching and the improved reversible transformation as shown in Fig. 1. In such example, we divide the SDs of secret blocks into three classes: SDs  $\{0, 1, 2, 3\}$  belong to the SD class 1



pixel is  $Over_{max}$  and the minimum underflow pixel is  $Un_{min}$ , we denote  $F = Over_{max} - 255$  (overflow) or  $F = -Un_{min}$  (underflow). Then  $\mathbf{P}'' = \{p''_1, p''_2, \dots, p''_n\}$  will be modified to  $\mathbf{P}''' = \{p'''_1, p'''_2, \dots, p'''_n\}$  as

$$\Delta p''' = \begin{cases} \Delta p'' - F, & \text{if } F \leq T \\ \Delta p'' - T, & \text{if } F > T \\ \Delta p'', & \text{No overflow/underflow pixel} \end{cases}, \quad (11)$$

where  $T$  is a threshold to make a tradeoff between the number of overflow/underflow pixels and the amplitude of bias from the mean of target block. The modified amplitudes  $F$ s need to be recorded, otherwise  $\Delta u$ 's cannot be restored by Eq. (9). It has been proved in the work [9] that setting  $T$  around 10 will perform well. For the sake of convenience to record  $F$ s, we set  $T = 8$  in this paper. If there is no pixel exceeding the range of 0-255 but exists the pixel equaling to 0 or 255 in the transformed block  $\mathbf{P}'''$ , we just set  $F = 0$ . If  $F = T = 8$ , this means after the modification there is no pixel exceeding the range of 0-255 in  $\mathbf{P}'''$ , but exists the pixel equaling to 0 or 255. In such case, the bias between secret block and transformed block can be still divisible by 8, so we just change it as  $F = 0$ . If  $F > 8$ , this means even modifying  $\mathbf{P}''$  to  $\mathbf{P}'''$  by the threshold  $T$ , there still exists the pixel exceeding the range of 0-255. In such case we record  $F = 8$ . Obviously,  $F$  is in the range of 0-8.

After getting the modified transformed block  $\mathbf{P}'''$ , we rotate it into one of the four directions  $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$  to fit the target block better as the strategy utilized in [8, 9]. Obviously, if the transformed block is perfectly smooth, no matter in which direction is rotated, the RMSE between the rotated version and the corresponding target block will not change. In other word, there is no need to rotate the transformed block having a small SD, i.e., the secret block having a small SD. Therefore, we only rotate the transformed block whose SD is larger than a given threshold  $SD_T$  to find the optimal direction, which is not considered in the previous works [8, 9].

By replacing each target block with the corresponding modified and rotated transformed block, we get a transformed image. Even we modify transformed block to reduce the number of overflow/underflow pixels, but overflow/underflow pixels may still exist in the yielded transformed image. To solve this problem, we just collect the modified transformed blocks for which the modified amplitude  $F = 8$  in order, since overflow/underflow pixels only exist in these blocks. Then we find the pixels which are no more than 0 or no less than 255 in these collected blocks, and truncate the overflow value to be 255 and the underflow value to be 0. To record the positions and residuals of truncations of these overflow/underflow pixels denoted by  $\mathbf{FP} = \{fp_1, fp_2, \dots, fp_m\}$ , we generate a sequence  $\mathbf{R} = \{r_1, r_2, \dots, r_m\}$  as

$$r_i = \begin{cases} fp_i - 255, & \text{if } fp_i \geq 255 \\ 0 - fp_i, & \text{if } fp_i \leq 0 \end{cases}, \quad 1 \leq i \leq m. \quad (12)$$

After truncating the overflow/underflow pixels of transformed image, we compress the parameters including  $MS$ s,  $F$ s,  $\mathbf{R}$ , the optimal directions and the SD class indexes of secret blocks, and then embed them

into the yielded transformed image with an RDH scheme.

### 3.2. The utilization of correlations within each color channel

#### 3.2.1. Predict the optimal direction

As mentioned above, after getting each transformed block we rotate it into one of the four directions  $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$  to find the optimal direction. The optimal direction is recorded for rotating the secret block in the anti-direction after recovering it from the transformed block. Indeed, the anti-direction can be well predicted based on the fact that there exists strong correlations among the neighboring pixels. After restoring each secret block, we rotate it to its original direction to yield secret image. However, if we rotate the restored secret block in a wrong direction, this may destroy the correlations, or the smoothness among the neighboring pixels at the edges of different blocks of the recovered image.

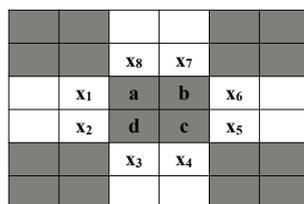


Figure 2: Rhombus pattern of secret blocks.

Taking the block size  $2 \times 2$  as example, we divide secret blocks into two parts labeled as dark and white respectively by using the rhombus pattern (see Fig. 2). Mention that only the optimal directions of dark blocks need prediction, and the optimal directions of white blocks are directly recorded. To predict the original direction (denoted by  $od$ ) of the dark block, we give a smoothness measurement denoted by  $smo$  to measure the smoothness of the edges between the dark block and the four nearby white blocks. The direction in which makes the minimum  $smo$  is deemed as the predicted original direction (denoted by  $\hat{od}$ ). Referring to Fig. 2,  $smo$  is calculated by

$$\begin{aligned}
 smo &= |x_1 - a| + |x_2 - d| + |x_3 - d| + |x_4 - c| \\
 &+ |x_5 - c| + |x_6 - b| + |x_7 - b| + |x_8 - a|.
 \end{aligned} \tag{13}$$

We use  $i = \{0, 1, 2, 3\}$  to represent the four directions  $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$  respectively, and  $smo_i$  is the smoothness measurement when rotating the dark block into the  $i$ th direction. Then  $\hat{od}$  is formulated as

$$\hat{od} = \arg \min_i \{smo_i\}, i \in \{0, 1, 2, 3\}. \tag{14}$$

There will be an error (denoted by  $e_{od}$ ) between the predicted direction and the original direction. Note that,  $e_{od}$  should be limited to the range of 0-3, so

$$e_{od} = \text{mod}(od - \hat{od}, 4), \quad (15)$$

where function  $\text{mod}(a, b)$  returns the modulus after division of  $a$  by  $b$ .

By the optimal directions of the white blocks (denoted by  $op_{WS}$ ), we rotate the white blocks back. With the help of the nearby white blocks, we can further rotate the dark secret blocks back by recording  $e_{ods}$ , which can be compressed efficiently.

### 3.2.2. Median-edge-detector

To compress  $MS$ s from three color channels of color secret image, we design the causal predictor to utilize the correlations within  $MS$ s for compression. Note that  $MS$ s from channels R and B will be compressed with  $MS$ s from channel G as reference, and the reason will be explained later. The median-edge-detector (MED) [19] based on half-enclosing causal  $MS$ s for prediction is adopted to compress  $MS$ s from channel G, and the decoder can inversely scan and process  $MS$ s to get the same prediction values with the help of several start  $MS$ s. Referring to Fig. 3, the prediction value of  $x$  is computed as

$$\hat{x} = \begin{cases} \min(u, v), & \text{if } w \geq \max(u, v) \\ \max(u, v), & \text{if } w < \min(u, v) \\ u + v - w, & \text{otherwise} \end{cases} . \quad (16)$$

Denote the prediction error (PE) of  $x$  by  $e_{MG}$ , where  $e_{MG} = x - \hat{x}$ . To recover  $MS$ s from channel G, we only need to record  $e_{MGS}$  (including several start values).

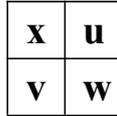


Figure 3: Context of an MS  $x$ .

Table 1: THE AVERAGE CORRELATION COEFFICIENTS AMONG THE THREE COLOR CHANNELS OF 1000 COLOR IMAGES.

$SD_{RG}$	$SD_{RB}$	$SD_{GB}$	$M_{RG}$	$M_{RB}$	$M_{GB}$
0.8071	0.6923	0.8305	0.9505	0.8374	0.9223

### 3.3. The utilization of correlations among three color channels

To verify the correlations among the three color channels of color images, we divide each channel of the color image into  $2 \times 2$  blocks and get the mean and SD of each block firstly. Then we calculate the correlation coefficients between the SDs from channels R and G denoted by  $SD_{RG}$ , from channels R and B denoted by  $SD_{RB}$ , from channels G and B denoted by  $SD_{GB}$ . Similarly, we calculate the correlation coefficients between the means from channels R and G denoted by  $M_{RG}$ , from channels R and B denoted by  $M_{RB}$ , from channels G and B denoted by  $M_{GB}$ . The average results of 1000 images from the BossBase image database [20] are shown in Table 1. From Table 1 we can see that after dividing the image into small blocks, the SDs and means of blocks from three color channels are correlated highly, which can be utilized efficiently to reduce AAI.

#### 3.3.1. Share the same compound index

Before transformation, we should pair each secret block with a target block having a similar SD, i.e., pair the secret block with a target block having the same compound index as Subsection 2.2 describes.

By the method [9], the image is divided into  $4 \times 4$  blocks and then these blocks are clustered into  $K$  classes according to their SDs by  $K$ -means. It has been proved in the work [9] that the quality of the created transformed image cannot be significantly improved by increasing  $K$  when  $K$  is larger than 6. In the proposed method, the block size is usually set as  $2 \times 2$ , and the distribution of SDs will be more sharper, so setting  $K = 6$  is proper in this paper.

It is easy to understand that the blocks from three color channels at the same position trend to share a similar complexity. Thus in the presented paper, the SD class indexes of blocks (determined by their SDs) from channels R, G and B at the same position are just deemed as the same. Both  $SD_{RG}$  and  $SD_{GB}$  are bigger than  $SD_{RB}$  (see Table 1), which shows that the complexity of the block from channel G can better represent the complexities of the blocks from R and B at the same position. Therefore, we assign each block from channel G a compound index, and the corresponding blocks at the same position from channels R, B will share the same compound index. The three-layer blocks are called a tile throughout the paper, and we will transform the secret tile to a paired target tile and gets the transformed tile. Experiments in Subsection 4.1 will show that such sharing is proper.

#### 3.3.2. Polynomial prediction

As mentioned before, the correlations among  $MS$ s are high. In this subsection, inspired by the adaptive predictor utilized in [18] we propose a new adaptive causal predictor which can provide good result for the compression of  $MS$ s from channels R and B.

Assuming the inputs  $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ , and the corresponding targets are  $\mathbf{Y} = \{y_1, y_2, \dots, y_n\}$ . A  $k$ th

degree polynomial is generalized to fit  $(x_i, y_i)$  for  $i = 1, 2, \dots, n$  as

$$f(x) = a_0 + a_1x + \dots + a_kx^k. \quad (17)$$

We can predict  $y_i$  based on  $x_i$  with Eq. (17), and the PE of  $y_i$  is

$$e_i = y_i - f(x_i) = y_i - (a_0 + a_1x_i + \dots + a_kx_i^k). \quad (18)$$

We search the polynomial coefficients which minimize the sum of the square of prediction errors with the weigh  $w_i$  as

$$\min\left\{\sum_{i=1}^n w_i e_i(\mathbf{A})^2\right\}. \quad (19)$$

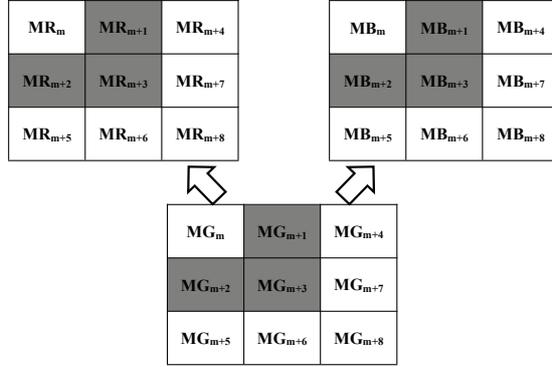


Figure 4: MSs from channel R, G, B.

The mean class indexes from channels R, G and B are denoted by  $MR$ ,  $MG$  and  $MB$  respectively in this subsection. Since  $M_{RG}$  and  $M_{GB}$  are higher than  $M_{RB}$  (see Table 1),  $MR$ s and  $MB$ s can be better predicted based on  $MG$ s by adopting the above polynomial prediction. As shown in Fig. 4, we predict the value of  $MR_m$  based on  $MG_m$  through exploring the correlations within the given  $n$  pairs of data  $(MG_i, MR_i)$ , where  $i = m + 1, m + 2, \dots, m + n$ . To explore the correlations within  $(MG_i, MR_i)$ , instead of using  $MR_i$  as the targets directly, we firstly calculate the differences between  $MG_i$  and  $MR_i$  denoted by  $\mathbf{D} = \{d_{m+1}, d_{m+2}, \dots, d_{m+n}\}$ , where  $d_i = MG_i - MR_i$ . Then the inputs  $\mathbf{X} = \{MG_{m+1}, MG_{m+2}, \dots, MG_{m+n}\}$  and the targets  $\mathbf{Y} = \{d_{m+1}, d_{m+2}, \dots, d_{m+n}\}$ . Since the correlations are higher for the closer regions of images, the weigh of  $e_i$  (i.e.,  $w_i$ ) is given as in inverse proportion to its squared distance from the predicted value  $\hat{d}_i$ . For example,  $\{w_{m+1}, w_{m+2}, w_{m+3}\} = \{1, 1, 1/2\}$ .

The prediction of  $d_m$  denoted by  $\hat{d}_m$  can be got by  $\hat{d}_m = f(MG_m)$  after finding the polynomial coefficients of Eq. (17). To get a better prediction, we limit  $\hat{d}_m$  in the range of  $d_{min}$  to  $d_{max}$ , where  $d_{min}$  is the minimum

value in  $\mathbf{D}$  and  $d_{max}$  is the maximum value in  $\mathbf{D}$ . Then the PE of  $d_m$  (denoted by  $e_d$ ) is formulated as

$$e_d = \begin{cases} d_m - d_{min}, & \text{if } \hat{d}_m \leq d_{min} \\ d_m - d_{max}, & \text{if } \hat{d}_m \geq d_{max} \\ d_m - \hat{d}_m, & \text{otherwise} \end{cases} \quad (20)$$

Obviously that, by recording  $e_{ds}$  (including several start values) which can be compressed efficiently, we can restore  $MRs$  based on  $MGs$ . In the same way,  $MBs$  can be also well compressed. For simplicity, we set  $k = 2$  and  $n = 3$  for the adopted polynomial prediction, and good effect of prediction can be also achieved.

### 3.3.3. Share the similar optimal directions

As mentioned in Subsection 3.3.1 that the blocks from R, B will share the same compound index as the block from G at the same position, and the secret tile (three-layer blocks) is transformed to the corresponding target tile. Due to the high correlations among the three channels of images, each block of the transformed tile trends to be rotated into the same direction to fit the target tile well.

Assuming that the optimal directions from channels R, G and B are denoted by  $op_R$  (represented by 0, 1, 2, 3),  $op_G$  and  $op_B$  respectively. In this paper,  $op_Gs$  are restored by recording  $op_Ws$  and  $e_{ods}$  as described in Subsection 3.2.1, and we use  $op_G$  as the prediction of  $op_R$  and  $op_B$ . Thus  $op_R$  and  $op_B$  can be restored by recording their PEs denoted by  $e_{RG}$  and  $e_{GB}$  respectively, where

$$\begin{cases} e_{RG} = \text{mod}(op_R - op_G, 4) \\ e_{GB} = \text{mod}(op_B - op_G, 4) \end{cases} \quad (21)$$

By recording  $e_{ops}$ , where  $e_{op} = \{e_{RG}, e_{GB}\}$ , we can rotate recovered secret blocks from channels R and B back with the help of  $op_Gs$ .

### 3.4. Algorithms of the proposed method

Based on the above discussion, the detailed processes of the presented RVT are described as follows:

#### Camouflage creation

1. Divide each color channel of the secret image and the target image into non-overlapped  $w \times h$  blocks, and assign a mean class index  $MS$  to each secret block according to its mean.
2. Compress  $MSs$  from channel G with MED as described in Subsection 3.2.2, and compress  $MSs$  from channels R and B with the proposed polynomial prediction as described in Subsection 3.3.2. Then record these  $MS$  PEs as parameters for recovery.
3. Cluster the secret tiles (three-layer blocks) into 6 classes according to their SDs from channel G as described in Subsection 2.2. And then cluster the target tiles into 6 classes according to their SDs from channel G, the secret SD classes' volumes and the scanning order. Assign an SD class index to each tile.

4. Assign a compound index  $i_j$  to each tile according to the SD class indexes and the scanning order, and pair each secret tile with a target tile having the same compound index.
5. For each pair of secret/target tiles, calculate  $\Delta u$  as Eq. (5), modify  $\Delta u$  as Eq. (6), and shift each secret block to the target block in the same channel with Eq. (7). Then adjust overflow/underflow blocks as Eq. (11) and record the modified amplitude  $F$ s for recovery.
6. Rotate the transformed block whose SD is larger than  $SD_T$  into the optimal direction. Predict the original directions from channel G as discussed in Subsection 3.2.1, and predict the optimal directions from channels R, B with the help of the optimal directions from channels G as discussed in Subsection 3.3.3. Then these direction PEs are recorded as parameters for recovery.
7. Replace each target block with the corresponding transformed block to generate the transformed image. Then find the overflow/underflow pixels in the modified transformed blocks for which the modified amplitude  $F = 8$  and truncate these overflow/underflow pixels. The positions and residuals of truncations are recorded in  $\mathbf{R}$  with Eq. (12).
8. Collect all the parameters including  $MS$  PEs, direction PEs,  $F$ s,  $\mathbf{R}$  and secret SD class indexes, and compress them separately with an entropy coder (such as arithmetic coder). Then encrypt these accessorial information with a standard encryption scheme controlled by key  $K$  and embed the encrypted sequence into the transformed image with an RDH scheme to create the ultimate camouflage image.

### Secret image recovery

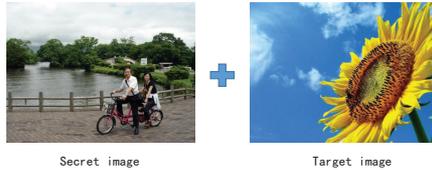
1. Extract the encrypted sequence and restore the transformed image with the RDH scheme. Then decrypt and decompress the sequence, and obtain the parameters including  $MS$  PEs, direction PEs,  $F$ s,  $\mathbf{R}$  and secret SD class indexes.
2. Divide each color channel of the restored transformed image into non-overlapped  $w \times h$  blocks. Then add the modified amplitudes  $F$ s and the truncated residuals  $\mathbf{R}$  to the pixels of transformed blocks, and assign a mean class index  $MT$  for each transformed block according to its mean.
3. From  $MS$  PEs, restore  $MS$ s of channels R, G and B.
4. Calculate the SDs of transformed blocks from channel G, and cluster the transformed tiles into 6 classes according to the calculated SDs.
5. Scan the transformed tiles in the raster order, and assign a compound index to each one. With secret SD class indexes, generate the secret compound indexes, according to which rearrange these transformed tiles.
6. For each transformed block, yield the secret block by Eq. (9) and Eq. (10) with  $MS$ s and  $MT$ s as parameters, and rotate the restored secret blocks back with the help of direction PEs.

## 4. Experimental results

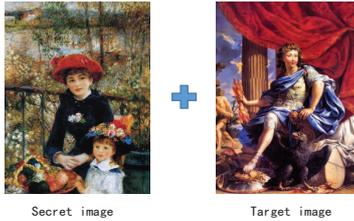
### 4.1. Settings on the proposed method

All the test images shown in the experiments are listed in [21], which are in the PNG format adopted by many camera equipments such as iPhone and IPad. In this subsection, three typical combinations (AAI is typical) as Fig. 5 shows are applied to discuss how to properly set on the proposed method.

As mentioned in Subsection 3.1, after getting each transformed block, we further rotate the transformed block whose SD is larger than a given threshold  $SD_T$  into the optimal direction to fit the corresponding target block well. A proper  $SD_T$  will reduce the number of transformed blocks to be rotated, thus reduce AAI for recording these optimal directions while not significantly affecting the quality of created transformed image. We search a proper  $SD_T$  with Fig. 5 as test images, and the results with  $2 \times 2$  as the block size are shown in Fig. 6. From Fig. 6 we see that the number of smooth blocks (corresponding SD is smaller than  $SD_T$ ) denoted by  $smo_{num}$  increases sharply with the increasing of SD when SD is smaller than 6, while the RMSE of created transformed image changes little. However, when SD is larger than 6, the RMSE of the created transformed image will increase rapidly but the number of smooth blocks increases slowly. Therefore,  $SD_T = 6$  is an appropriate threshold.



(a) Example 1



(b) Example 2



(c) Example 3

Figure 5: (a) (b) (c) are three typical combinations.

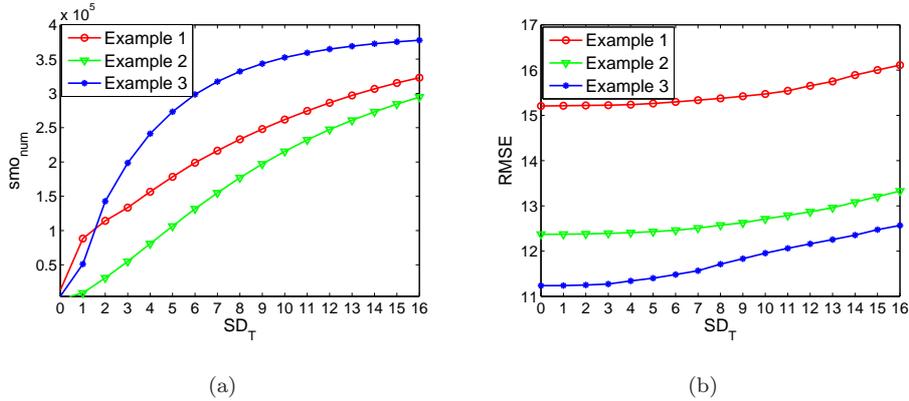


Figure 6: (a) The influence of  $SD_T$  on the number of the smooth blocks. (b) The influence of  $SD_T$  on the RMSE of the transformed image.

The blocks from channels R, B will share the same compound index as the block from channel G at the same position as mentioned in Subsection 3.3.1. Now, we give experiments with  $2 \times 2$  as the block size to show that such sharing is proper. Two groups of experiments with Fig. 5 as test images are carried out. In the first group, we cluster the blocks of three color channels respectively, and each block from different channels has its independent compound index just as the method adopted in the work [9]. In the second group, the three blocks at the same position share the same compound index as the block from channel G. The experimental results are listed in Table 2, which show that the difference between the created transformed images of the above two groups is subtle, while AAI for recording secret SD class indexes in the second group only accounts for one-third of that in the first group.

Table 2: THE RMSES OF THE TRANSFORMED IMAGES IN TWO GROUPS.

Groups	Example 1	Example 2	Example 3
The first group	9.8788	8.2104	7.8326
The second group	10.2336	8.4126	8.0220

As mentioned before, the smaller block size will improve the quality of transformed image while causing more accessorial information, and the distortion between the ultimate camouflage image and the corresponding target image is caused by following two factors. One is the transformation to generate transformed image, and the other is embedding the accessorial information into transformed image. In the following experiment with [16] as the adopted RDH scheme, a proper block size is sought to improve the quality of ultimate camouflage image as much as possible. The experimental results of Fig. 5 by setting different block sizes are given in Fig. 7. From Fig. 7 we can see that the size  $2 \times 2$  performs the best for Example 1 and Example 3, and with the block size becoming larger, the quality of transformed image and corresponding camouflage image both will decrease. For Example 2, the size  $2 \times 2$  nearly performs the best as the size  $4 \times 4$  does.

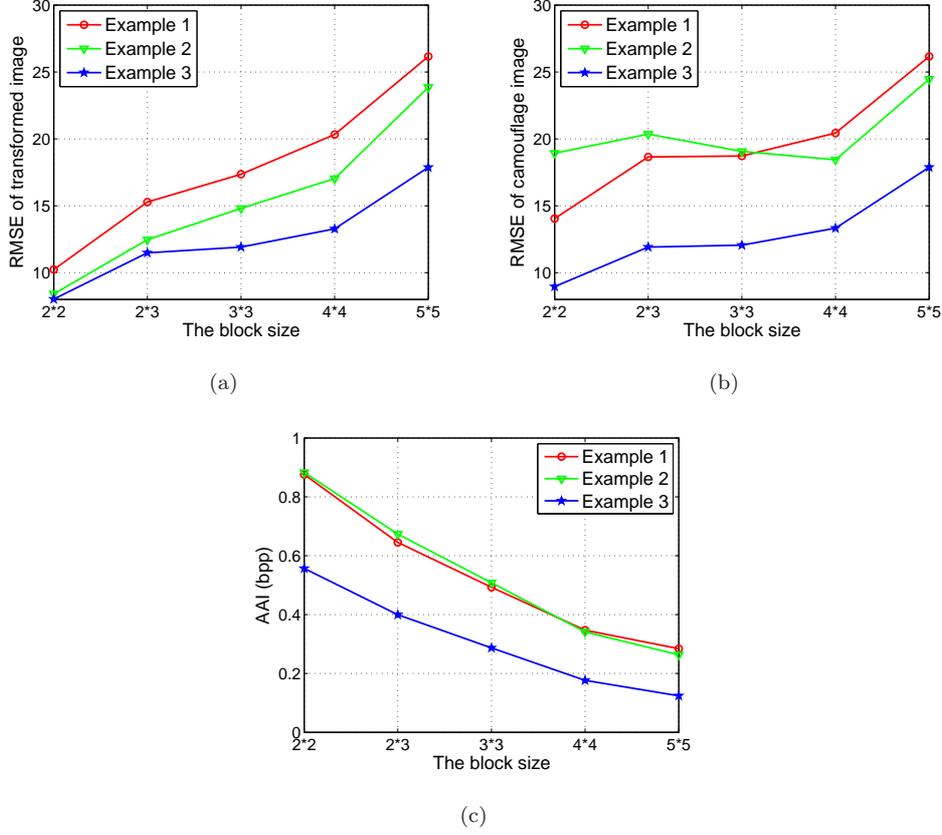


Figure 7: (a) The RMSE of the transformed images. (b) The RMSE of the camouflage images. (c) The amount of the accessorial information.

Thus we think setting the block size as  $2 \times 2$  is usually proper.

However, Example 2 in Fig. 7 also shows that the block size  $2 \times 2$  is not always the optimal size, sometimes a larger block size can yield the better camouflage image. The reason is that there may exist some abnormal combinations which will cause the relatively larger AAI than the general combinations, or that the adopted RDH scheme may cause relatively greater distortion for the transformed images generated from these abnormal combinations. To solve this problem, we adaptively select a proper block size for each combination to balance the aforementioned two factors causing distortion. Assuming that AAI reflected by bits per pixel (bpp) is denoted by  $\rho$ , the RMSE between the transformed image and the target image is denoted by  $D_1$ , and the RMSE between the ultimate camouflage image and the target image is denoted by  $D_2$ . Under the condition that the block size is  $2 \times 2$ :

- If  $D_2 - D_1 > D_1$ , the distortion caused by the adopted RDH scheme is greater than that caused by the transformation;
- If  $\rho > 1$ , we think AAI is relatively large when setting the block size as  $2 \times 2$ .

For each given combination, we do the transformation with the block size as  $2 \times 2$  firstly. But as long as one of the above hypotheses occurs, we enlarge the block size  $2 \times 2$  to be  $4 \times 4$  to reduce AAI, and we only need additional one bit to record the block size as  $2 \times 2$  or  $4 \times 4$ . When the block size is set as  $4 \times 4$  which is the same as that for the method [9], the quality of transformed image created by the proposed method will be similar with that created by the method [9], but AAI is much reduced.

The experiments are carried with MATLAB-R2014a. The test machine is lenovo PC with 4130 CPU @ 3.40GHz and 4.00 GB RAM. The runtimes for generating transformed image with size  $768 \times 1024$  is around 52 seconds by setting  $2 \times 2$  block and 15 seconds by setting  $4 \times 4$  block. There is still great space for speed optimization.

#### 4.2. Comparisons with previous methods

To make fair comparisons with the previous works [8, 9], we adopt the same RDH scheme proposed in [16] to embed the accessorial information into the transformed image for the proposed method and the previous methods [8, 9]. The block size  $8 \times 8$  and  $4 \times 4$  usually perform best for the methods [8] and [9] respectively, so we set the block sizes as  $8 \times 8$  and  $4 \times 4$  respectively for the methods [8] and [9]. In addition to RMSE, the mean structural similarity (SSIM) [22] is also adopted to appraise the similarity between the camouflage image and the corresponding target image, and the window size for computing SSIM is set to be  $8 \times 8$ .

##### 4.2.1. One-round transformation

Now, we give a comparative experiment as Fig. 8 shows, from which we can see that the quality of camouflage image (Fig. 8(e)) generated by the proposed method outperforms those generated by Lee *et al.*'s method [8] (Fig. 8(c)) and the method [9] (Fig. 8(d)) a lot. The jigsaw puzzle effect is further reduced by the presented method as shown in the zoom-out images (Fig. 8(g)-(i)). Note that both the presented method and the method [9] can losslessly recover the secret image while Lee *et al.*'s method [8] can not. The detailed data of RMSE and SSIM of the created camouflage images by the above three methods is shown in Table 3.

Table 3: THE SIMILARITIES BETWEEN CAMOUFLAGE IMAGES AND CORRESPONDING TARGET IMAGES.

Camouflage image	RMSE	SSIM
Fig.8(c)	28.85	0.4567
Fig.8(d)	21.22	0.6093
Fig.8(e)	12.51	0.7361

To verify the feasibility of the presented method, we test it on 100 pairs of images, which are randomly depicted from the BossBase image database [20] and sampled down to  $1024 \times 1024$ . The average results of

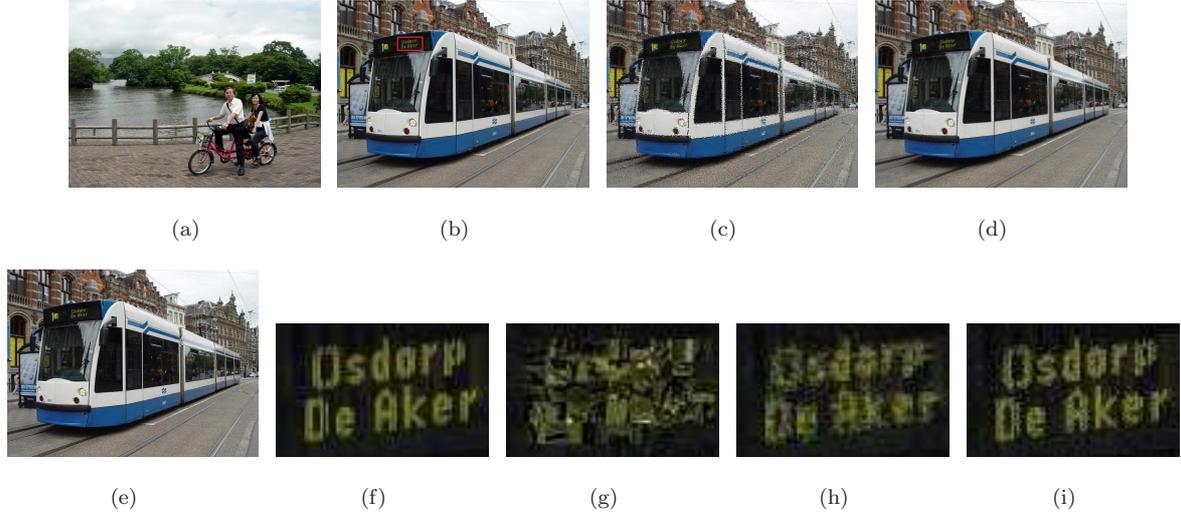


Figure 8: (a) Secret image. (b) Target image. (c) Camouflage image created by Lee *et al.*'s method. (d) Camouflage image created by the method [9]. (e) Camouflage image created by the presented method. (f) Zoom-out image of red square region from image (b). (g) Zoom-out image of the same region from image (c). (h) Zoom-out image of the same region from image (d). (i) Zoom-out image of the same region from image (e).

Table 4: THE AVERAGE RESULTS ON 100 PAIRS OF TEST IMAGES.

Method	RMSE of transformed image	RMSE of camouflage image	AAI
Lee <i>et al.</i> 's method	19.23	22.16	0.5309
The method [9]	13.29	16.03	0.6133
Presented method	9.82	12.98	0.5331

the proposed method and the previous methods [8, 9] are listed in Table 4. From Table 4 we can see that, the presented method outperforms the previous methods [8, 9], because we can set the smaller block size through utilizing the correlations within color images to reduce AAI for recovering each secret block. Even we set most of the block sizes as  $2 \times 2$  in the presented method (the block size is enlarged to  $4 \times 4$  in 7 combinations of the above 100 combinations), AAI is still similar with that of Lee *et al.*'s method [8] with the block size as  $8 \times 8$ , and less than the method [9] with the block size as  $4 \times 4$ .

#### 4.2.2. Comparisons with LSB method

Indeed, the direct way to realize RVT is compressing secret image and embedding the compressed bitstream into the target image to create the camouflage image with the method of Least Significant Bit (LSB) replacement. If the secret image is smooth enough and can be compressed efficiently, its compressed bitstream will be small. Then RVT can be well realized by such LSB method, and the camouflage image will be also good. However, even sometimes the quality of camouflage image generated from LSB method is acceptable, but the correlations within the generated camouflage image are relatively weak because of

the irregular bitstream, which will cause difficulty when further doing RDH on such camouflage image. As mentioned earlier, doing RDH on camouflage image is a novel client-free framework [11] for RDH-EI. While by the presented method the pixels of camouflage image are the shifted pixels of secret image after RDH, thus the correlations can be kept, which is very necessary for RDH. What is more, if the texture of secret image is complex, the amount of compressed bitstream will be large. In such case, LSB method will cause great distortion for the camouflage image, while the proposed RVT can achieve a much better result.

PNG format adopted by iPhone and iPad etc is the most widely used lossless format. Firstly, we embed the PNG file stream of secret image into the target image with the method of LSB to generate the camouflage image. The selected secret image (Fig. 9(a)) has rich texture information, and the amount of secret PNG file streams reaches 6.42 bits for each target pixel. We believe that even the best traditional data hiding method cannot well accommodate so much file streams. So Fig. 9(c) created by LSB method is much poorer than Fig. 9(d) created by the presented method.

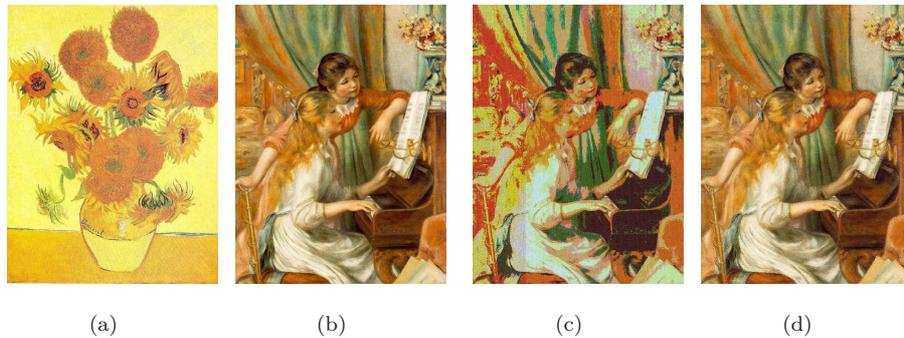


Figure 9: (a) Secret image. (b) Target image. (c) Camouflage image created by LSB method with RMSE= 39.26 with respect to Fig. 9(b). (d) Camouflage image created by the proposed method with RMSE= 25.60 with respect to Fig. 9(b).

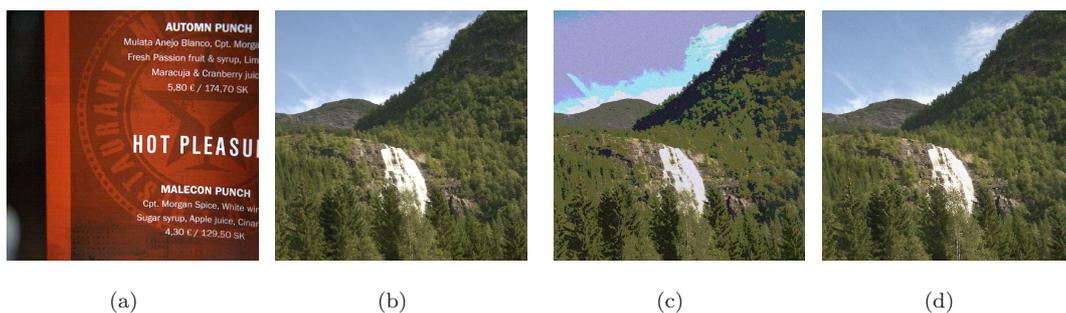


Figure 10: (a) Secret image. (b) Target image. (c) Camouflage image created by LSB method with RMSE= 22.01 with respect to Fig. 10(b). (d) Camouflage image created by the proposed method with RMSE= 17.79 with respect to Fig. 10(b).

JPEG2000 is a more powerful compressor. We can firstly compress secret image with JPEG2000 compressor, and then embed the compressed bitstream into the target image with the method of LSB to generate the camouflage image. However, even with JPEG2000 some images still cannot be well compressed. For

example, the compressed bitstream of Fig. 10(a) with JPEG2000 reaches 5.87 bits for each target pixel. Therefore, Fig. 10(c) created by LSB method is still poorer than Fig. 10(d) created by the presented method.

As mentioned before, doing RDH in camouflage image is a novel framework for RDH in encrypted image. A sharper host histogram will accommodate a higher capacity under the same distortion constraint, which will be more suitable for the further RDH. Half of PEs got with rhombus prediction utilized in [16] of the camouflage images are shown in Fig. 11, from which we can see that the PE histogram from the camouflage image created by the proposed method is much sharper than that created by LSB method. The reason is that LSB method will destroy the correlations within the camouflage image a lot while the presented method will not. For example, by these histogram-shift based RDH schemes [23] which embed messages into the top bin of host histogram and shift the other bins, the capacities on half PEs are 26985 bits, 39808 bits for Fig. 9(c), Fig. 9(d) respectively, and 61377 bits, 92212 bits for Fig. 10(c), Fig. 10(d) respectively. Therefore, the presented RVT is preferred to do the client-free RDH-EI scheme.

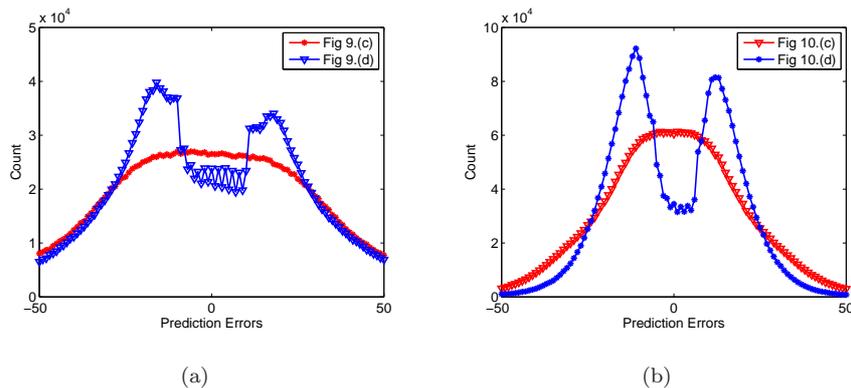


Figure 11: (a) Average PEs from Fig. 9. (b) Average PEs from Fig. 10(d).

#### 4.2.3. Multi-round transformation

The RVT scheme can be used to realize multi-round transformation, i.e., transform the camouflage image to another target image. Multi-round transformation is not allowed by Lee *et al.*'s method due to its irreversibility. In this subsection, we carry out multi-round transformation by adopting the presented method, and make comparisons with the method [9]. Note that when doing multi-round transformation, the block size is set as  $4 \times 4$  for the presented method the same as that for the method [9].

Two series of experiments on multi-round transformation are shown in Fig. 12 and Fig. 13. In 2-round transformation, the sender firstly transforms the secret image  $A$  to the target image  $B$ , getting the one-layered camouflage image  $B_H$  and  $B_P$  with the method [9] and the proposed method respectively. Then the sender transforms  $B_H$  and  $B_P$  to their target image  $C$  to get the two-layered camouflage image  $C_H$  and  $C_P$

respectively. At the receiver side, the receiver will firstly restore the one-layered camouflage image from the two-layered camouflage image, and then restore the secret image from the one-layered camouflage image.

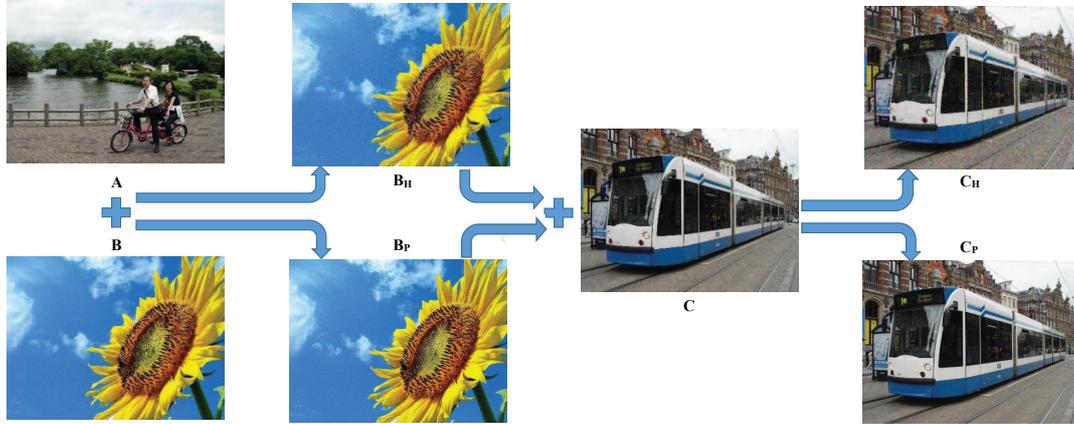


Figure 12:  $A$  is a secret image.  $B$  is a target image.  $B_H$  is the camouflage image created by the method [9] with RMSE= 24.01 with respect to  $B$ .  $B_P$  is the camouflage image created by the presented method with RMSE= 20.63 with respect to  $B$ .  $C$  is another target image.  $C_H$  is the camouflage image created by the method [9] with RMSE= 31.58 with respect to  $C$ .  $C_P$  is the camouflage image created by the presented method with RMSE= 21.13 with respect to  $C$ .

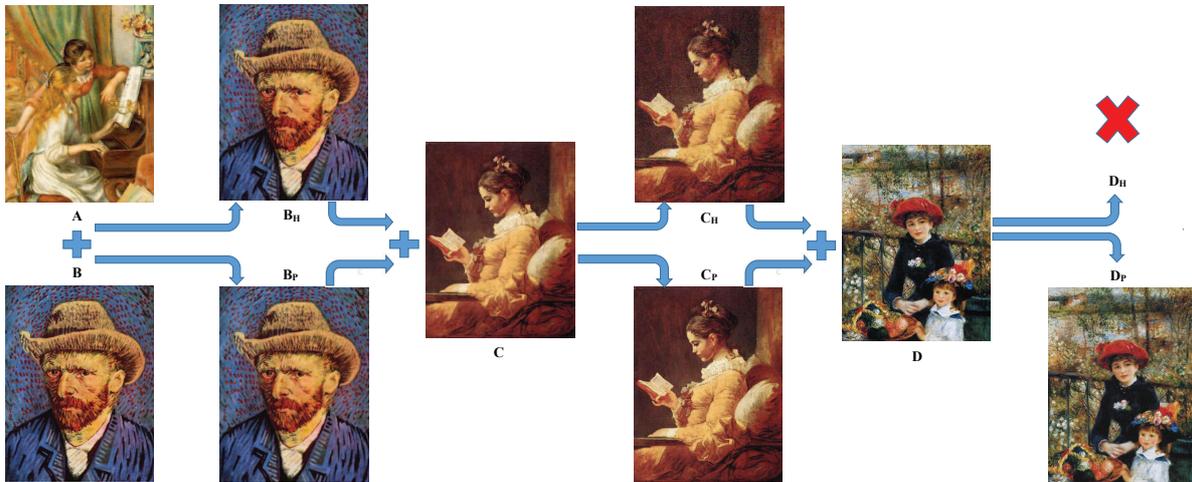


Figure 13:  $A$  is a secret image.  $B$  is a target image.  $B_H$  is the camouflage image created by the method [9] with RMSE= 14.45 with respect to  $B$ .  $B_P$  is the camouflage image created by the presented method with RMSE= 12.97 with respect to  $B$ .  $C$  is another target image.  $C_H$  is the camouflage image created by the method [9] with RMSE= 22.52 with respect to  $C$ .  $C_P$  is the camouflage image created by the presented method with RMSE= 16.22 with respect to  $C$ .  $D$  is the third target image.  $D_H$  cannot be created by the method [9].  $D_P$  is the camouflage image created by the presented method with RMSE= 28.65 with respect to  $D$ .

Although in this subsection the block size for the presented method is the same as that for the method [9], the camouflage image  $B_P$  created by the presented method is better than the camouflage image  $B_H$

created by the method [9]. Indeed, the corresponding transformed images of  $B_P$  and  $B_H$  have a similar quality, but  $B_P$  is better than  $B_H$  due to that by the presented method the accessorial information is less and the less accessorial information will cause the less distortion for the transformed image. For example, in Fig. 12 the corresponding transformed images of  $B_P$  and  $B_H$  have  $RMSE = 20.22$  and  $RMSE = 19.84$  respectively, but AAI for creating  $B_P$  is 0.3472 bpp and for creating  $B_H$  is 0.7706 bpp. Thus  $B_P$  is better than  $B_H$  with  $RMSE = 20.63$  and  $RMSE = 24.01$  respectively.

Note that, although there are two target images  $B$  and  $C$  in the process of 2-round transformation, we completely embed the accessorial information into the shifted pixels of image  $A$  with an RDH scheme. Because the capacity of RDH in one image is finite, in the second round the RDH will destroy the quality of image more seriously. Thus the two-layered camouflage image  $C_P$  created by the presented method is much better than the two-layered camouflage image  $C_H$  created by the method [9]. For the same reason, in Fig. 13 the 3-round transformation is not allowed by the method [9], but can be realized by the presented method.

## 5. Conclusion and discussion

In this paper, we propose a new RVT technique for color images by exploring and utilizing the correlations within color images. Thus AAI for recovering each secret block is largely reduced, by which we improve the quality of camouflage image a lot through setting the smaller block size.

In the scheme of RVT, RDH scheme is required to embed the accessorial information into the transformed image. As we can see from the above examples, sometimes the RDH scheme will destroy the quality of transformed image greatly, thus a good RDH method is desired. Since the generated transformed image is in an RGB color format, the adopted RDH scheme [16] for gray images is not suitable. Although some RDH techniques for color images [24, 25] have been proposed, their capacities are limited for the RVT scheme. In the future, we will try to design new RDH technique for the RVT scheme, by which continue to improve the visual quality of the ultimate camouflage image.

## References

- [1] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 234–239, Dec 2012.
- [2] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, pp. 1–13, 2014.
- [3] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Trans. Information Forensics and Security*, vol. 10, no. 9, pp. 1905–1917, 2015.
- [4] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

- [5] Fridrich J, Kodovsky J. “Rich models for steganalysis of digital images,” *IEEE Trans. Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.
- [6] T. Denemark, V. Sedighi, V. Holub, R. Cograume, and J. Fridrich, “Selection-channel-aware rich model for steganalysis of digital images,” in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 48–53, IEEE, 2014.
- [7] I.-J. Lai and W.-H. Tsai, “Secret-fragment-visible mosaic image—a new computer art and its application to information hiding,” *IEEE Trans. Information Forensics and Security*, vol. 6, no. 3, pp. 936–945, 2011.
- [8] 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 Trans. Circuits Syst. & Video Technol.*, vol. 24, no. 4, pp. 695–703, 2014.
- [9] D. Hou, W. Zhang, and N. Yu, “Image camouflage by reversible image transformation,” *Journal of Visual Communication and Image Representation*, vol. 40, Part A, pp. 225–236, 2016.
- [10] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, 2001.
- [11] W. Zhang, H. Wang, D. Hou, and N. Yu, “Reversible data hiding in encrypted images by reversible image transformation,” *IEEE Trans. Multimedia*, vol. 18, pp. 1469–1479, 2016.
- [12] X. Zhang, “Reversible data hiding in encrypted image,” *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.
- [13] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, “Reversible data hiding in encrypted images by reserving room before encryption,” *IEEE Trans. Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.
- [14] Liao X., Shu C., “Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels”, *Journal of Visual Communication and Image Representation*, 2015, 28: 21-27.
- [15] Pan Z., Hu S., Ma X., Wang L., “Reversible data hiding based on local histogram shifting with multilayer embedding”, *Journal of Visual Communication and Image Representation*, 2015, 31: 64-74.
- [16] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, “Reversible watermarking algorithm using sorting and prediction,” *IEEE Trans. Circuits Syst. & Video Technol.*, vol. 19, no. 7, pp. 989–999, 2009.
- [17] Lu, T. C., Tseng, C. Y., Deng, K. M., “Reversible data hiding using local edge sensing prediction methods and adaptive thresholds,” *Signal Processing*, 104, 152-166, 2014.
- [18] I.-C. Dragoi and D. Coltuc, “Local-prediction-based difference expansion reversible watermarking,” *IEEE Trans. Image Processing*, vol. 23, no. 4, pp. 1779–1790, 2014.
- [19] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The loco-i lossless image compression algorithm: principles and standardization into jpeg-ls,” *IEEE Trans. Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.
- [20] BossBase image database, [Online]. Available: <http://agents.fel.cvut.cz/stegodata/RAWs/>.
- [21] Related images of the experiments, [Online]. Available: [http://people.cs.nctu.edu.tw/%7Eylllee/ylllee&whtsai\\_sfv.html](http://people.cs.nctu.edu.tw/%7Eylllee/ylllee&whtsai_sfv.html).
- [22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [23] Z. Ni, Y. Shi, N. Ansari, and S. Wei, “Reversible Data Hiding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, 2006.
- [24] J. Li, X. Li, and B. Yang, “Reversible data hiding scheme for color image based on prediction-error expansion and cross-channel correlation,” *Signal Processing*, vol. 93, no. 9, pp. 2748–2758, 2013.
- [25] B. Ou, X. Li, Y. Zhao, and R. Ni, “Efficient color image reversible data hiding based on channel-dependent payload partition and adaptive embedding,” *Signal Processing*, vol. 108, pp. 642–657, 2015.