J. Vis. Commun. Image R. 40 (2016) 225–236

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Image camouflage by reversible image transformation ☆,☆☆

Dongdong Hou, Weiming Zhang *, Nenghai Yu

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

ARTICLE INFO

Article history:
Received 23 September 2015
Revised 14 May 2016
Accepted 20 June 2016
Available online 30 June 2016

Keywords:
Image camouflage
Image transformation
Image encryption
Reversible data hiding

ABSTRACT

A new reversible image transformation technique is proposed, which not only improves the visual quality of the camouflage image created by transforming a secret image to a freely-selected target image, but also can restore the secret image without any loss. Effective clustering algorithm is utilized to reduce the information for recording block indexes that is vital for restoring the secret image. Therefore, the transformation can be made between the blocks with relatively small size, thus greatly improving the visual quality of the camouflage image. The root mean square error of camouflage image was reduced by 6 in most cases compared with the previous method. Since the proposed technique is reversible, we can further realize two-round transformation by transforming the camouflage image to another target image and thus hide two images into only one.

1. Introduction

Nowadays, outsourcing photos to cloud or sharing photos through social media is more and more popular, which at the same time make it challenging to protect the privacy of photos' owners. For instance, recently many private photos of Hollywood actress leaked from iCloud [1].

There are two common approaches, encryption and data hiding, to protect image contents from leakage. Although encryption solves the privacy problem in a certain extent, but the messy codes of ciphertext with special form are easy to cause the attention of attackers who will plan to breakout the accounts of encryption users.

Data hiding technology embeds message into covers such as the 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. The most secure approach of data hiding is to embed the message while minimizing a suitably defined distortion, such as methods proposed in [2–4], which have strong ability to resist detection, but can only achieve a relatively small payload, less than 1 bit per pixel (bpp). Traditional data hiding method is suitable for embedding a small message into a large cover, e.g., an image. However, in the applications of image outsourcing or sharing, the message itself is just an image. Therefore, we need large capacity data hiding (LCDH) methods to hide images. Although, LCDH is hard to resist detection of strong steganalysis [5–7], it will be very useful in the applications of privacy protection of photos.

LCDH can be used for "secret image sharing" by hiding one image into several other images [8,9]. Typical secret image sharing model is based on the well known (t, n) threshold scheme originated from Blakley [10] and Shamir [11], which generates n shadow images and any t $(t \leqslant n)$ of n shadow images can be used to reconstruct the secret image. However, in such scheme, t is larger than 1, that is, we need more than one image to reconstruct the secret image. How to hide one image into another one with the same size is a more challenging problem, which we call "image transformation".

As shown in Fig. 1(a), by image transformation, the sender can transform a secret image A to a target image B, getting a camouflage image B' which is similar with the target image B. From B', the recipient can reconstruct a A', and usually A' is a good estimation of A. If A can be losslessly reconstructed from B', i.e., A' is equal to A, we call the scheme reversible image transformation, just as Fig. 1(b) shows. Image transformation technique can be viewed as a special kind of data hiding method, which embeds image A into B.

Lai et al. [12] propose an image transformation method, which selects a target image similar to the secret image in a database, then replaces each block of the target image by a similar block of the secret image and embeds the map between secret blocks and target blocks, finally generates a camouflage image as the

☆ This paper has been recommended for acceptance by M.T. Sun.

☆☆ This work was supported in part by the Natural Science Foundation of China under Grant 61170234 and Grant 60803155, and by the Strategic and Piloted Project of CAS under Grant XDA06030601.

* Corresponding authors.
E-mail addresses: houdd@mail.ustc.edu.cn (D. Hou), zhangwm@ustc.edu.cn (W. Zhang), ynh@ustc.edu.cn (N. Yu).

http://dx.doi.org/10.1016/j.jvcir.2016.06.018
1047-3203/

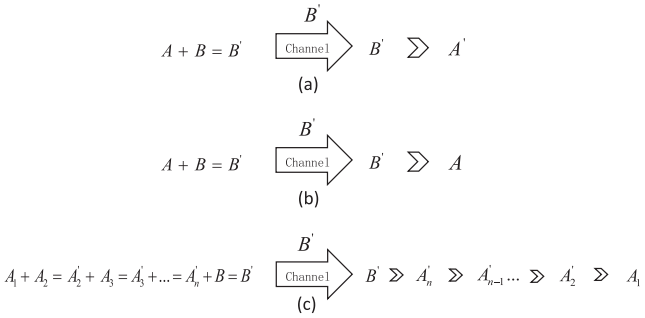


Fig. 1. Diagram of image transformation. $A, A_1, A_2, \ldots, A_n$ are secret images; B is a target image. (a) Nearly-reversible image transformation. (b) Reversible image transformation. (c) Multi-round transformation.

camouflage of the secret image. A greedy search is used to find the most similar block. Although Lai et al.'s method is reversible, it is only suitable for a target image similar with the secret image, and the visual quality of the created camouflage image is not so good.

Lee et al. [13] improve Lai et al.'s method by transforming the secret image to a freely-selected target image without the need of a database. In Lee et al.'s method, each block of the secret image is transformed to a block of the target image with a nearly-reversible color transformation [14], and then the accessorial information for restoring secret image, such as transformation parameters, indexes of block, is embedded into the transformed blocks, getting the ultimate camouflage image. Lee et al.'s method can transform a secret image to a freely-selected target image, and greatly improves the visual quality of the camouflage image. However, by Lee et al.'s method, the secret image cannot be losslessly reconstructed because the transformation is not reversible.

Reversible image transformation is important especially for the applications of medical images and military images. Moreover, the reversibility enables us to realize multi-round transformation as shown in Fig. 1(c), in which the image A_1 is transformed to A_2 getting A_2', and then A_2' is transformed to A_3 getting A_3', and so on. Finally we hide n secret images, $A_1, \ldots, A_n$, into one target image B getting B'. From B', the receiver can in turn reconstruct $A_n', A_{n-1}', \ldots, A_2'$ and A_1. Note that the receiver only completely restores A_1 and gets similar versions for other secret images. Without the reversibility, the accessorial information will be destroyed in the process of restoration, and thus only one round transformation is possible. In the multi-round transformation, $A_2, \ldots, A_n$ are secret images and target images at the same time. Therefore, to realize multi-round transformation, the target image must be freely-selected, because secret images may be arbitrary.

Such multi-round transformation can be used for different purposes such as follows:

- Hiding several secret images, $A_1, A_2, \ldots, A_n$, into one target image B.
- If only A_1 is the secret image, deeply hiding A_1 into several images by using different keys in each round and thus realizing multi-layered camouflage.
- Compressing $n+1$ images, $A_1, A_2, \ldots, A_n$ and B, into one image B'. Such compression is transparent, because B' is similar to B, and thus B' can be viewed as the envelop of the compressed files.

In this paper, we propose a reversible image transformation for freely-selected target images. To keep the reversibility, we transform the secret block to the target block only by shifting the mean of pixels in the secret block. A clustering algorithm is used to divide all blocks into K classes according to their standard deviations (SDs). The clustering can efficiently reduce the information for recording the indexes of block, which enables us to set small block size and thus greatly improves the visual quality of the created camouflage images. To accelerate the speed of transformation, the parallel framework proposed in [15,16] may be useful. Because the proposed method is reversible and is suitable for arbitrary target images, we can further realize two-round transformation.

The rest of the 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. Previous arts

The proposed method is an improvement of Lee et al.'s method [13], which we will briefly introduce firstly. Both the proposed method and Lee et al.'s method do transformation for channel R, G, B of a color image separately, so we just take the transformation on gray images (one channel) as an example in Sections 2 and 3. In Lee et al.'s method, the secret image and the target image are divided into N non-overlapping blocks with the same size, which are called tiles. The secret tiles are sorted into a sequence $\mathbf{B}_i$ $(1 \leqslant i \leqslant N)$, and the target tiles are sorted into another sequence $\mathbf{T}_i$ $(1 \leqslant i \leqslant N)$ according to the SD of the pixels in each tile. And then the ith secret tile is transformed to the ith target tile with the following near-reversible transformation.

Let secret tile $\mathbf{B}$ be a set of pixels such that $\mathbf{B} = \{p_1, p_2, \ldots, p_n\}$, and the corresponding target tile $\mathbf{T} = \{p_1', p_2', \ldots, p_n'\}$. Firstly calculate the mean and SD of each tile.

$$u = \frac{1}{n}\sum_{i=1}^{n} p_i, \quad u' = \frac{1}{n}\sum_{i=1}^{n} p_i'. \tag{1}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - u)^2}, \quad \sigma' = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i' - u')^2}. \tag{2}$$

A new set of pixels $\mathbf{T}' = \{p_1'', p_2'', \ldots, p_n''\}$ is generated by calculating as

$$p_i'' = q(p_i - u) + u', \tag{3}$$

where $q = \sigma'/\sigma$. Obviously $\mathbf{T}'$ has the same mean and variance as the target tile $\mathbf{T}$. Replace each $\mathbf{T}$ with the corresponding $\mathbf{T}'$, we get a transformed image (Before embedding accessorial information.) similar with the target image. At the receiver side, the original pixel p_i can be recovered by

$$p_i = 1/q(p_i'' - u') + u. \tag{4}$$

To recover the original pixels, the parameter q, u and u' must be embedded into the transformed image and sent to the receiver. To embed the real number q, it is represented as a number in the range of 0.1–12.8 with 7 bits. These parameters are embedded into the transformed image with a reversible data hiding (RDH) scheme [17] and the yielded camouflage image will be sent to the receiver. The RDH scheme enables the receiver to losslessly reconstruct the transformed image after extracting these parameters and then recover the secret image from the transformed image with the help of the parameters. Note that p_i'' yielded with Eq. (3) is also a real number which must be truncated to be an integer in the range of 0–255, so the original pixel value cannot be recovered exactly by Eq. (4). That is why Lee et al.'s method is not reversible.

Besides the parameters of transformation, the sender must embed the indexes of secret tiles into the transformed image, according to which the receiver rearranges the restored tiles to get the secret image. With the smaller tile size, we can get a trans-

formed image with better quality. However, the smaller tile size is, the more accessorial information should be embedded into the transformed image which may introduce two problems. One is that the RDH scheme cannot reach an enough capacity to accommodate these accessorial information. The other one is that the more information been embedded into the transformed image, the worse quality the camouflage image will result in.

In the present paper, we will improve Lee et al.'s method in two aspects:

- Modify the transformation, (3) and (4), to be reversible;
- Reduce the information for recording tile indexes and thus improve the quality of camouflage image by setting the smaller tile size.

3. Proposed method

3.1. Reversible transformation

To realize reversibility, we simplify the transformation (3) by abandoning q and get the following shift transformation.

$$p_i'' = p_i + u' - u, \tag{5}$$

where $u' - u$ is the difference between the means of target tile and secret tile. We want to shift each pixel value of secret tile by amplitude $u' - u$ and thus the transformed tile has the same mean with the corresponding target tile. Because the pixel value p_i'' should be an integer, to keep the transformation reversible, we round the difference to be the closest integer.

$$\Delta u = round(u' - u), \tag{6}$$

and shift the pixel value by Δu, namely, each p_i'' is got by

$$p_i'' = p_i + \Delta u. \tag{7}$$

Note that the pixel value p_i'' should be an integer between 0 and 255, so the transformation (7) may result in some overflow/underflow pixel values. If overflow/underflow problem occurs, we reduce the number of overflow/underflow pixels by modifying Δu as follows. In the transformed tile obtained by Eq. (7), assuming that the maximum overflow pixel value is Ov_{max} and the absolute value of the minimum underflow pixel is Un_{min}, Δu is modified as Eqs. (8) and (9). When $\Delta u > 0$:

$$\Delta u = \begin{cases} \Delta u + 255 - Ov_{max}, & \text{if } (Ov_{max} - 255) < T \\ \Delta u - T, & \text{if } (Ov_{max} - 255) \geqslant T \end{cases}, \tag{8}$$

and when $\Delta u < 0$:

$$\Delta u = \begin{cases} \Delta u + Un_{min}, & \text{if } Un_{min} < T \\ \Delta u + T, & \text{if } Un_{min} \geqslant T \end{cases}, \tag{9}$$

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 tile.

The parameter Δu will be embedded into the transformed image by an RDH scheme. To reduce the bits for recording Δu, we quantize it by

$$\Delta u = \begin{cases} 8 \times round\left(\frac{\Delta u}{8}\right), & \text{if } \Delta u > 0 \\ 8 \times floor\left(\frac{\Delta u}{8}\right) + 4, & \text{if } \Delta u < 0 \end{cases}, \tag{10}$$

where *floor* means rounding the element to the nearest integer not larger than it. With the quantized Δu, we use Eq. (7) to update the transformation.

To send the quantized Δu, we only need to record $\Delta u' = \frac{|\Delta u|}{4}$, by which the range of Δu is quartered with even number meaning $\Delta u \geqslant 0$ and odd number meaning $\Delta u < 0$. Obviously, we can recover the quantized Δu from $\Delta u'$. In fact, the values of $\Delta u'$ concentrate in a small range close to zero which can be efficiently compressed by an entropy coder.

Even with the modified Δu, the overflow/underflow problem can still occur. To solve this problem, we find the pixels of transformed tile which are no more than 0 or no less than 255, and truncate the overflow value to be 255, the underflow value to be 0. To record the positions and residuals of truncations, we generate a sequence $\mathbf{R} = \{r_1, r_2, \ldots, r_n\}$ such that

$$r_i = \begin{cases} p_i'' - 255, & \text{if } p_i'' \geqslant 255 \\ 0 - p_i'', & \text{if } p_i'' \leqslant 0 \end{cases}. \tag{11}$$

The values in $\mathbf{R}$ concentrate in a small range close to zero which can be efficiently compressed.

As proposed in [13], we further rotate the truncated tile into one of the four directions 0°, 90°, 180° or 270°, and make the final transformed tile more similar to the target tile. We rotate the tile into the optimal direction which makes the minimum root mean square error (RMSE) between the rotated version and the corresponding target tile.

After shifting transformation, truncation and rotation, we get a new tile $\mathbf{T}'$. By replacing each $\mathbf{T}$ in the target image with $\mathbf{T}'$ we get a transformed image. After that, we compress the parameters include $\Delta u'$s, rotation directions, and $\mathbf{R}$, then embed them into the transformed image with an RDH scheme.

3.2. Tiles matching by non-uniform clustering

Before transformation, we should find the proper target tile for each secret tile. In Lee et al.'s method, the tiles of secret image and target image are sorted according to their SDs respectively, and then each secret tile is transformed to a corresponding target tile in turn according to the order. To recover the secret image, the index information of secret tiles should be embedded into the transformed image. However, this method is not suitable for small tile sizes or big images. For instance, if we divide a 1024×768 image into 4×4 tiles, we need 16 bits to record the index of each tile. Besides these index information, we should also embed other parameters into the transformed image, so it is hard to keep a good visual quality for the ultimate camouflage image.

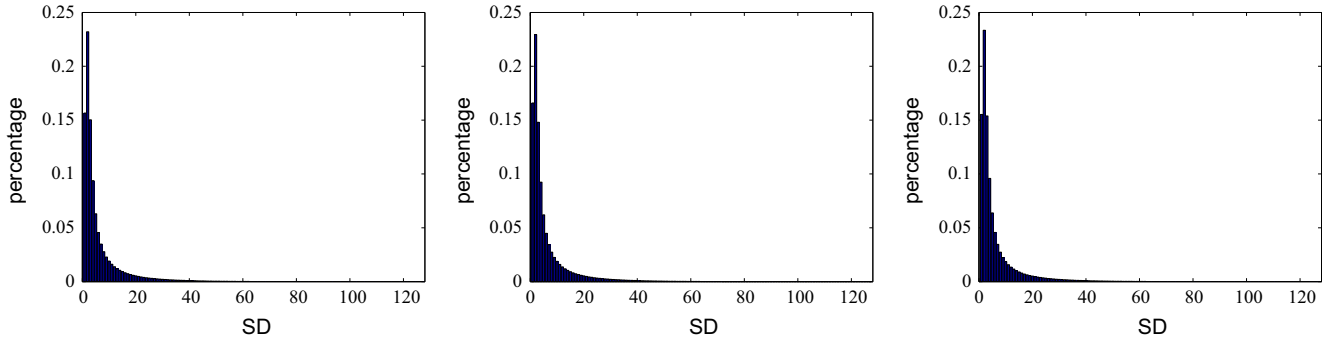
In this section, we use a clustering algorithm to reduce the information for recording the tile indexes. We tested 6000 nature images to observe the distributions of SDs of 4×4 tiles. The distribution of average SDs of these images is shown in Fig. 2. It can be seen that most of SDs of 4×4 tiles concentrate in a small range close to zero and the frequency fast drops with the increase of the SD value. Since most of SDs are similar, there is no need to match the secret tiles with the target tiles strictly according to the order of SDs, which is also the reason that we abandon q in the transformation Eq. (3).

We divide the secret and the target image into non-overlapping 4×4 tiles and calculate the SD of each tile. In one image, we will scan the tile in the raster order, i.e., from left to right and from top to bottom.

Since most of SDs are similar, the tiles with close SDs are deemed as one class. We want to transform the secret tile to the target tile with a close SD value. To do that we only need to transform the secret tile to the target tile in the same class.

We firstly cluster all the SDs (and thus the corresponding tiles) of the secret image into K classes by classic clustering algorithm such as K-means. The K classes is sorted to ensure that the SDs in the ith class is smaller than in the jth class for $1 \leqslant i < j \leqslant K$.

Next, we cluster the tiles of target image based on the classes' volumes of secret image and the scanning order, and set each class to have the same volume as the corresponding class of the secret



(a) Distribution of the average SDs from channel R.

(b) Distribution of the average SDs from channel G.

(c) Distribution of the average SDs from channel B.

Fig. 2. Distribution of the average SDs of 4×4 tile from nature images.

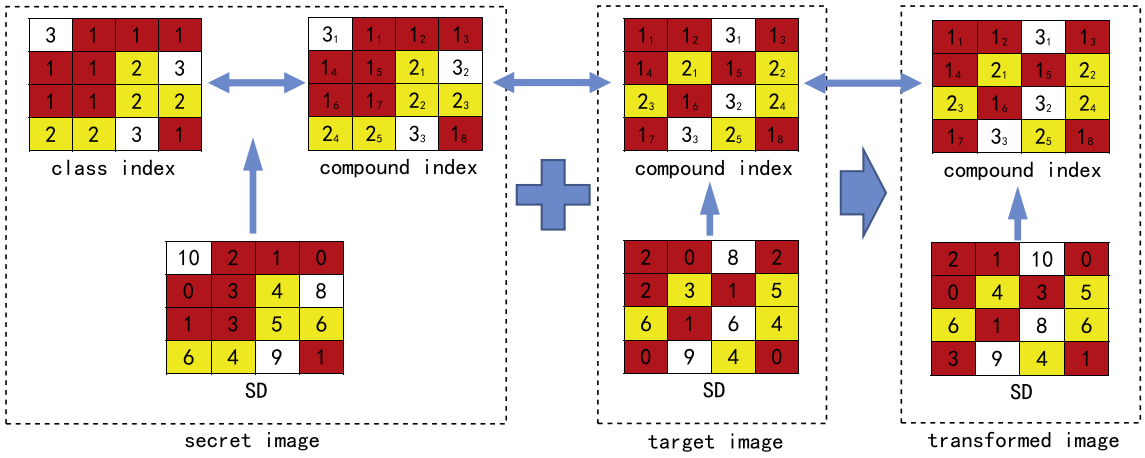


Fig. 3. An example of tiles matching by non-uniform clustering.

Table 1
The one-to-one correspondence among tile indexes, class indexes and compound indexes.

Class index	3	1	1	1	1	1	2	3	1	1	2	2	2	2	3	1
Compound index	3_1	1_1	1_2	1_3	1_4	1_5	2_1	3_2	1_6	1_7	2_2	2_3	2_4	2_5	3_3	1_8
Tile index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

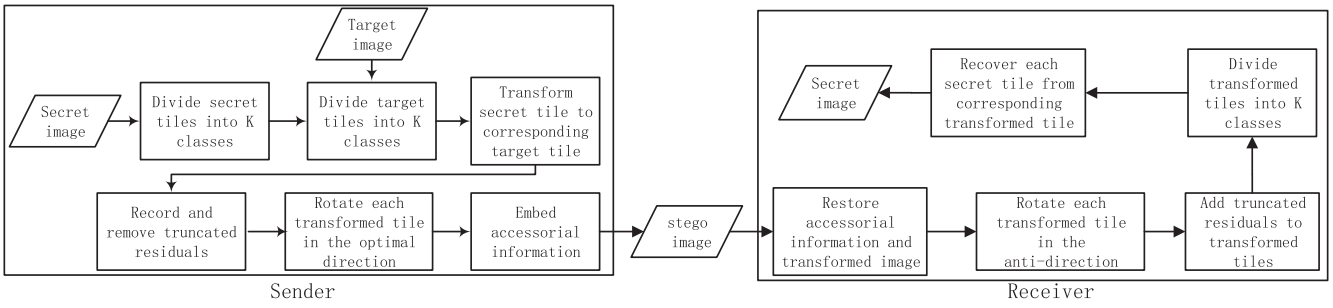


Fig. 4. Flow diagram of the proposed method.

image. Assuming that the ith class in the secret image includes n_i tiles for $1 \leqslant i \leqslant K$. We divide the first n_1 tiles of target image with the smallest SDs into the first class and the next n_2 tiles with the smallest SDs of the rest of the tiles into the second class, and so on, until all the tiles of target image are divided. Now, we divide both secret tiles and target tiles into K classes.

In each class, we sort the tiles according to the scanning order and assign a compound index to each tile such that the jth tile belonging to the ith class is labeled as the i_jth tile $(1 \leqslant i \leqslant K, 1 \leqslant j \leqslant n_i)$. By the compound indexes, we get the one to one map between the secret tiles and the target tiles. We will transform the i_jth secret tile to the i_jth target tile and replace the

i_jth target tile with the corresponding transformed tile, which will yield a transformed image.

Because shift and rotation do not change the SD of each secret tile, so the transformed tiles have the same SD values with the corresponding secret tiles. Thus, the receiver can cluster the transformed tiles into the same K classes as those of the secret image and assign a class index to each tile. Furthermore, by scanning the tiles in the raster order, the receiver can assign each transformed tile a compound index which is equal to that of the target tile at the same position. In other words, the classes' scatter pattern of the transformed image is the same with that of the target image, because the transformed tiles are arranged according to the compound indexes of the target image. To recover the secret image from the transformed image, we need to record compound indexes of the secret image according to the scanning order. To do that we only need to record the class indexes of secret tiles in the scanning order, which is illustrated in the following example.

A simple example of the proposed tiles matching method is shown in Fig. 3, in which we divide the SDs of secret tiles into three class: SDs $(0,1,2,3)$ belong to the class 1, SDs $(4,5,6)$ belong to the class 2, SDs $(8,9,10)$ belong to the class 3. We label the first class as "red", the second class as "yellow" and the third class as "white". In the target image, although two tiles have the same SD value "6", the first one is assigned to class 2 but second one is assigned to the class 3, because class 2 only can include 5 tiles as determined by the classes of the secret image. For each image, by scanning the class index in the raster order, we can define the compound index as shown in the secret image. For instance, according to the scanning order, the first secret tile is the first one of class 3 that is labeled as 3_1, and the second secret tile is the first one of class 1 that is labeled as 1_1.

By the compound indexes, we establish one-to-one map between the secret tiles and the target tiles and thus transformation is made between the paired tiles, that is, transforming the i_jth secret tile to the i_jth target tile. For example, we transform the first secret tile to the third target tile, the second secret tile to the first target tile, and replace the target tiles with the corresponding transformed tiles. Finally we generate a transformed image whose classes have the same scatter pattern as those of target image. The i_jth secret tile will be restored from the i_jth transformed tile. Therefore, to get the secret image, we should record the index of the i_jth secret tile. The information on such tile index can be established by the class indexes as shown in Table 1.

Note that we cluster the secret tiles according to their SDs, and the frequency of SDs which nearly satisfies an exponential distribution as shown in Fig. 2, thus the class indexes also have a sharp distribution. Therefore, the clustering is non-uniform and the class indexes of secret tiles can be efficiently compressed by an entropy coder and then embedded into the transformed image by an RDH scheme.

3.3. Algorithms of proposed method

The flow diagram of the proposed method is given as Fig. 4. The detailed algorithms for camouflage image creation and secret image recovery are described as follows.

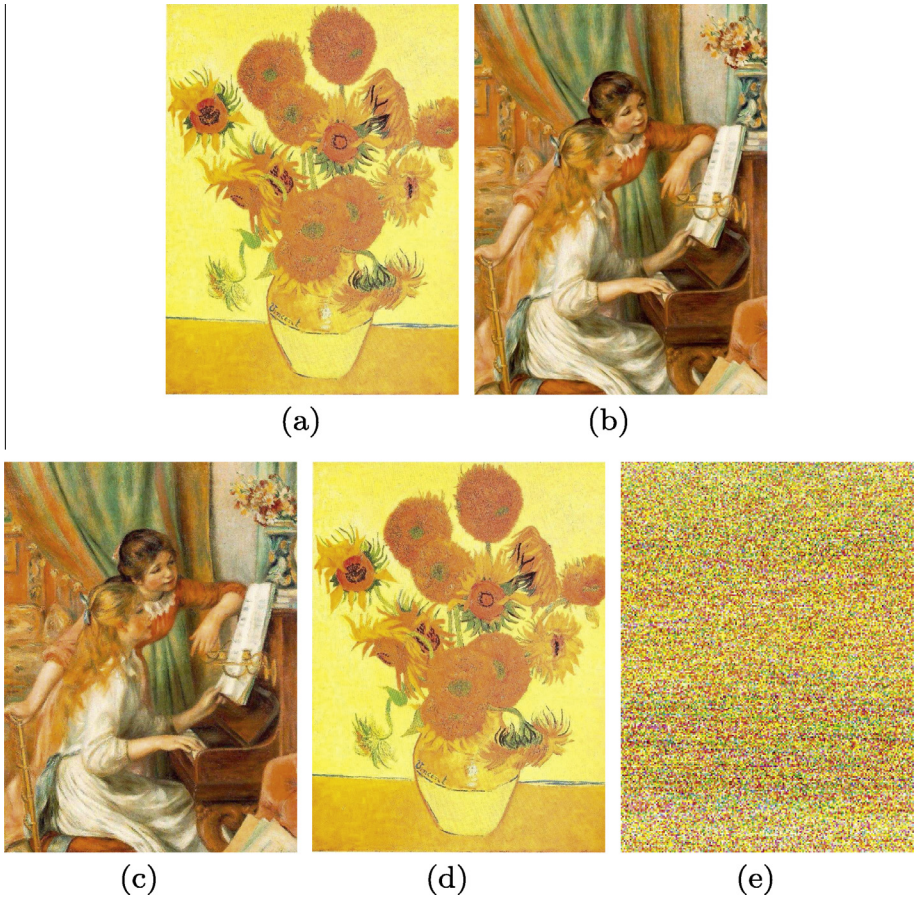
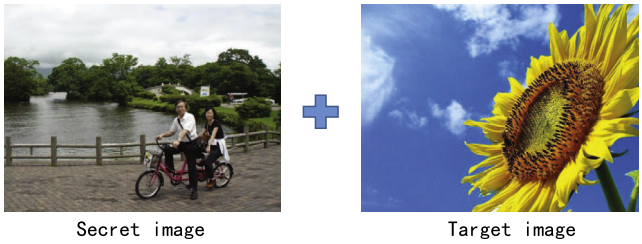
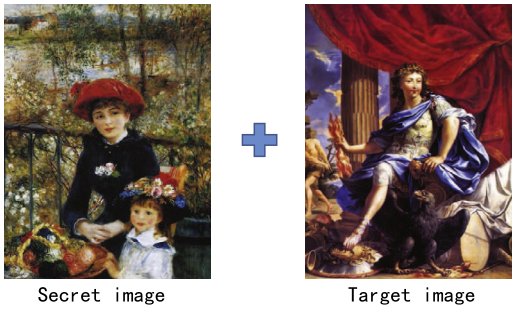


Fig. 5. (a) Secret image. (b) Target image. (c) Camouflage image created from (a) and (b). (d) Recovered secret image using right key. (e) Recovered secret image using a wrong key.



(a) Example 1



(b) Example 2

(c) Example 3

Fig. 6. (a)–(c) Are three different combinations used to create transformed images (before embedding accessorial message).

Camouflage image creation

- **Input:** A secret image, target image and a secret key P.
- **Output:** A camouflage image.
- **Step 1**. Divide each color channel c ($c = r, g$, and b) of secret image and target image into non-overlapping 4×4 tiles, and calculate the SD of each tile. For each color channel, do step (2)–(7).
- **Step 2**. Cluster the secret tiles into K classes according to their SDs. And then cluster the target tiles into K classes according to the SDs, the classes' volumes of secret image and the scanning order. Assign a class index to each tile.
- **Step 3**. Assign a compound index i_j to each tile according to the class indexes and the scanning order, and pair the secret tile with a target tile having the same compound index.
- **Step 4**. For each pair of secret/target tiles, calculate the mean of secret tile u_c and the mean of target tile u'_c. Calculate $\Delta u = u'_c - u_c$, and modify Δu with Eqs. (6), (8)–(10). By the modified Δu, shift each value of the secret tile with Eq. (7), and record $\Delta u' = |\Delta u|/4$ as the parameter for restoring Δu.
- **Step 5**. Find the pixels of each transformed tile which are no more than 0 or no less than 255, and truncate the overflow/underflow pixel values. Record the positions and residuals of truncations with Eq. (11).
- **Step 6**. Rotate each transformed tile into the optimal one of four directions 0°, 90°, 180° or 270°, and record the rotation direction as a parameter.
- **Step 7**. Replace each target tile with the corresponding transformed tile.
- **Step 8**. Combine three color channels to generate the transformed image.
- **Step 9**. For each secret tile, the parameters of transformation include the class' index, $\Delta u'$, rotation direction. Collect all the parameters and all the truncated residuals, and compress them separately with an entropy coder.
- **Step 10**. Encrypt the accessorial information with a standard encryption scheme controlled by key P. Embed the encrypted sequence into the transformed image with an RDH scheme to create the ultimate camouflage image.

Secret image recovery

- **Input:** A camouflage image and the secret key P.
- **Output:** The secret image S.
- **Step 1**. Extract the encrypted sequence and restore the transformed image with the RDH scheme.
- **Step 2**. Decrypt and decompress the sequence, and obtain the truncated residuals and the parameters of transformation, including the class' indexes of secret image, $\Delta u'$s, rotation directions.
- **Step 3**. Divide each color channel of transformed image into non-overlapping 4×4 tiles. Rotate each transformed tile in the anti-direction according to the recorded direction in the parameters.
- **Step 4**. Add the truncated residuals to the pixel values of the transformed tiles. For each color channel, do step (5)–(7).

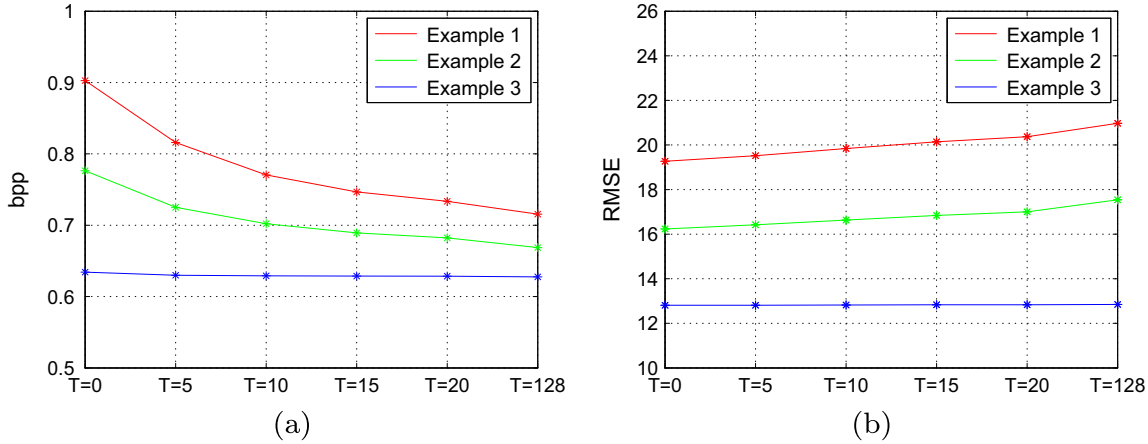


Fig. 7. (a) The influence of T on the amount of accessorial information. (b) The influence of T on the RMSEs of transformed images.

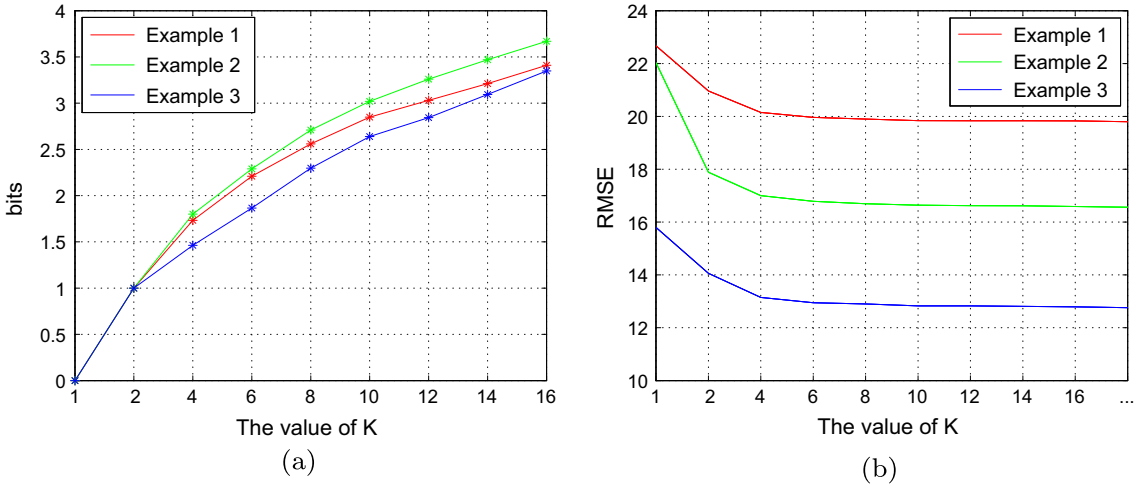


Fig. 8. (a) The influence of K on the amount of the information for recording per tile index. (b) The influence of K on the RMSEs of transformed images.

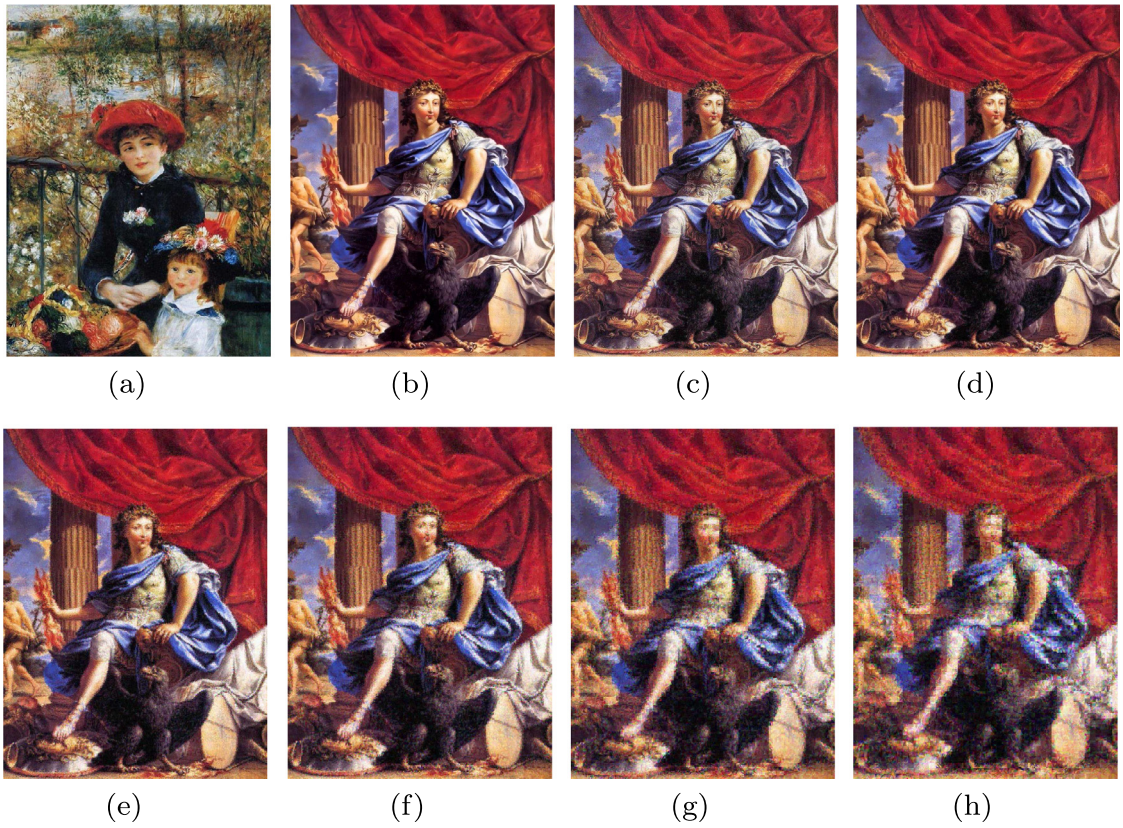


Fig. 9. (a) Secret image. (b) Target image. (c) Camouflage image by setting 3×3 block. (d) Camouflage image by setting 4×4 block. (e) Camouflage image by setting 6×6 block. (f) Camouflage image by setting 8×8 block. (g) Camouflage image by setting 12×12 block. (h) Camouflage image by setting 16×16 block.

- **Step 5**. Calculate the SD of each transformed tile, and cluster the tiles into K classes according to their SDs.
- **Step 6**. Scan the transformed tiles in the raster order, and assign a compound index to each one. With the class' indexes of secret image, generate the compound indexes of the secret tiles, according to which rearrange the transformed tiles.
- **Step 7**. For each transformed tile, restore Δu from $\Delta u'$ and subtract Δu from each color value, yielding the restored secret tile.
- **Step 8**. Combine three color channels to generate the secret image.

4. Experimental results

4.1. Settings on the proposed method

In the proposed method, we embed the accessorial information into the transformed image to get the ultimate camouflage image with an RDH scheme. To realise two-round transformation, we need an RDH technique with high embedding capacity. Many RDH techniques [18–20] can achieve high embedding capacity while preserving the image quality, and we adopt the method proposed in [18].

Table 2
The results by setting the different block size.

Block size	Transformed image (RMSE)	Camouflage image (RMSE)	MAI (bpp)
Fig. 9(c) (3 × 3)	13.35	39.09	1.19
Fig. 9(d) (4 × 4)	16.67	21.76	0.7013
Fig. 9(e) (6 × 6)	21.18	22.92	0.3501
Fig. 9(f) (8 × 8)	24.39	24.75	0.2331
Fig. 9(g) (12 × 12)	29.14	29.29	0.1580
Fig. 9(h) (16 × 16)	32.28	32.36	0.1358

To keep the security, the accessorial information should be compressed and then encrypted before being embedded into the transformed image. We propose to use Huffman coder to compress the accessorial information and use AES to encrypt the compressed sequence. Fig. 5 illustrates the security of the proposed method. The camouflage image (Fig. 5(c)) is similar with the target image (Fig. 5(b)) which becomes a good camouflage. Even recognizing the camouflage, without the correct key, the attacker cannot decrypt the accessorial information and thus can not get any information of the secret image as shown in Fig. 5(e).

To reduce the number of overflow/underflow pixels, we adjust Δu within a small range by Eqs. (8) and (9), which will make bias between the means of the transformed tile and the corresponding target tile. A threshold T is used to limit the amplitude of the mean's bias. We divide the secret tiles into 10 classes firstly, namely setting $K = 10$, and then do a series of experiments to observe the influence of T on the number of overflow/underflow pixels and the quality of transformed image. Three typical experiments are shown in Figs. 6 and 7. The influence of T on the number of overflow/underflow pixels is shown in Fig. 7(a) reflected by bpp, and the influence on the RMSE between the transformed image and the corresponding target image is shown in Fig. 7(b). The value of T nearly has no effect on Experiment 3, just because there are few overflow or underflow problems in the process of the tile transformation. For Experiment 1 and 2, when T is smaller than 10, the number of overflow/underflow pixels decreases quickly with the increase of T, while the effect of transformed image changes little. It seems that we can still increase the threshold T to reduce the accessorial information, and also could keep good effect of the transformed image because the large residual amplitude is a rare case. However, too big T will greatly influence the quality of some special transformed tiles with large residual amplitude. To avoid the influence of the abnormally large mean's bias, we still set $T = 10$.

As mentioned above, we divide the SDs into K classes. Now, we will give examples to show that, which K is appropriate. We still use the combinations in Fig. 6 as test images, and the results by setting the different K are shown in Fig. 8. Fig. 8 shows that the amount of the information needed for recording per tile index increases with K while RMSE of the transformed image decreases with K. When K is larger than 6, we cannot significant reduce RMSE by increasing K. So it seems that $K = 6$ is a good setting. However, there may exist some abnormal tiles that cannot be paired properly when K is small, so we set $K = 10$. When $K = 10$, we only need about 3 bits to record per tile index, which is much reduced. That is because the SDs concentrate in a small range close to zero as shown in Fig. 2.

Now we give an experiment as Fig. 9 shows by setting the different block sizes. From Fig. 9 and Table 2 we can see that when the block size is set to be 3 × 3, the amount of the accessorial information will reach 1.19 bpp, which is relatively large for RDH scheme. So Fig. 9(c) is poor although the corresponding transformed image has a good quality. From Fig. 9(d)–(h) we can see that with the block size becoming larger, the amount of the accessorial information (MAI) will be reduced, but the quality of the camouflage image and the corresponding transformed image is decreasing.

4.2. Comparisons with previous methods

To make comparisons, we adopt the same RDH scheme [18] to embed the accessorial information in both the proposed method

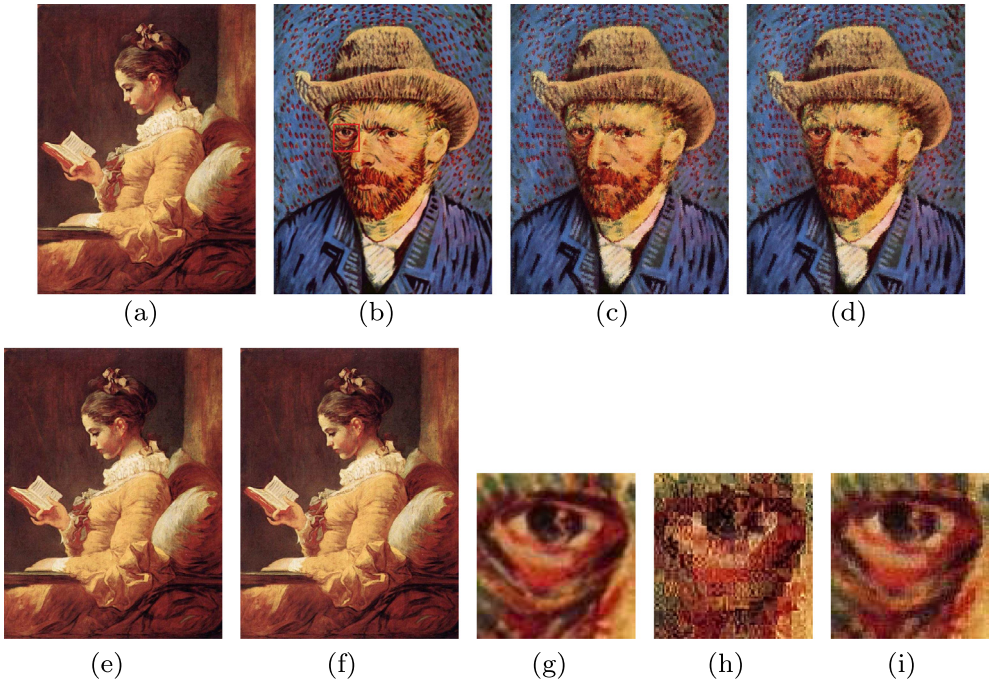


Fig. 10. (a) Secret image. (b) Target image. (c) Camouflage image created by Lee et al.'s method. (d) Camouflage image created by the proposed method. (e) Recovered secret image by Lee et al.'s method with RMSE = 0.948 with respect to secret image. (f) Recovered secret image by the proposed method with RMSE = 0 with respect to secret image. (g) Zoom-out image of red square region from image (b). (h) Zoom-out image of the same region from image (c). (i) Zoom-out image of the same region from image (d). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
The similarity between camouflage images and corresponding target images.

Camouflage image	RMSE	SSIM
Fig. 10(c)	22.38	0.3562
Fig. 10(d)	13.85	0.6535

Table 4
The similarity between camouflage images and corresponding target images.

Camouflage image	RMSE	SSIM
Fig. 11(c)	28.85	0.4567
Fig. 11(d)	21.22	0.6093

and Lee et al.'s method [13]. Two series of experiments are used to compare the proposed method with the previous method. In addition to RMSE, we also adopt the mean structural similarity (SSIM) [21] to appraise the similarity between two images. The tile size of 8×8 performs best in Lee et al.'s method, so we set the tile size as 8×8 for Lee et al.'s method, but 4×4 for the proposed method. The window size for computing the SSIM is set to be the same as 8×8.

From Fig. 10 and Table 3, we can see that the quality of the camouflage image generated by the proposed method outperforms that generated by Lee et al.'s method in both RMSE and SSIM. The proposed method can significantly reduce the jigsaw puzzle effect as shown in the zoom-out images Fig. 10(h) and (i). The reason is that the proposed method can use the smaller tile size thanking for the non-uniform clustering. Importantly, the proposed method can losslessly recover the secret image while Lee et al.'s method cannot.

In this experiment as shown in Fig. 11, we use a target image (Fig. 11(b)) that is irrelative with the secret image. Although Lee et al.'s method can use freely-selected image, there is visible distortion in the camouflage image (Fig. 11(c)), while the camouflage image (Fig. 11(d)) generated by the proposed method can still keep good transparency. The detailed data of RMSE and SSIM of the created camouflage images by the above two methods is shown in Table 4.

We need to embed the accessorial message for recovering the secret image. As mentioned above, through simplifying the transformation and adopting the non-uniform clustering, we reduce the accessorial information greatly, which enables us to set the smaller tile size and thus improves the visual quality of the camouflage images. The amount of the accessorial information needed for

Table 5
The amount of accessorial information of Lee et al.'s method and the proposed method.

Combinations	Lee's method for 8×8 tile	Lee's method for 4×4 tile	Proposed method for 4×4tile
Fig. 5	0.5045	1.8609	0.7019
Fig. 6(a)	0.7253	2.0463	0.7706
Fig. 6(b)	0.5495	1.8785	0.7021
Fig. 6(c)	0.4808	1.8371	0.6290
Fig. 10	0.4830	1.8259	0.6307
Fig. 11	0.5576	1.8702	0.6596

Table 6
The average results on 100 pairs of test images.

Method	RMSE of transformed image	RMSE of camouflage image	MAI
Proposed method	13.29	16.03	0.6133
Lee et al.'s method	19.23	22.16	0.5309

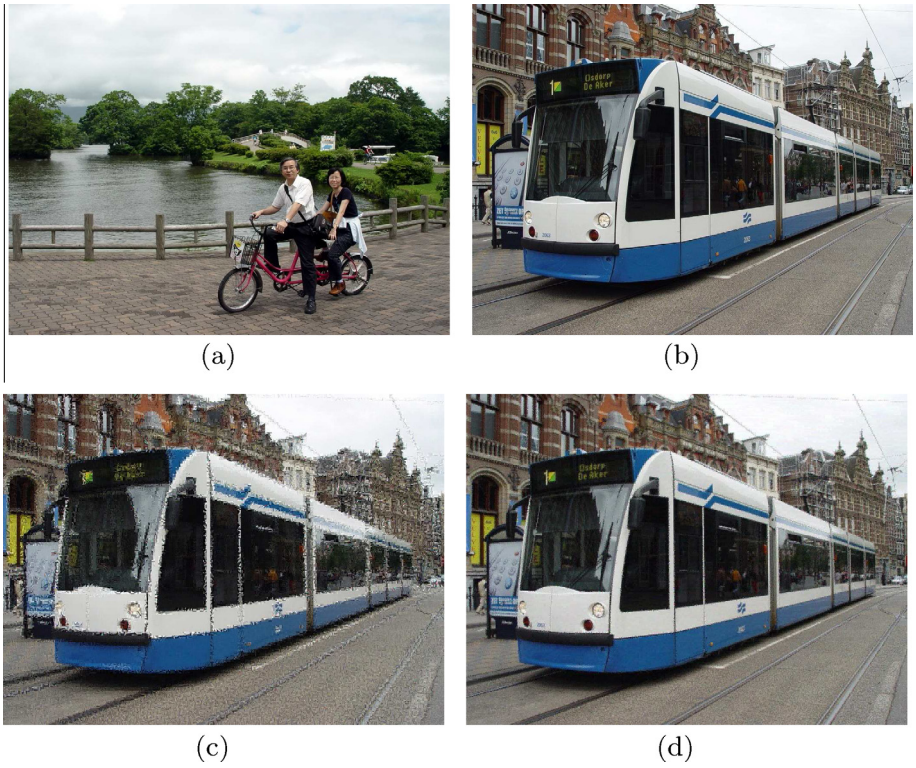


Fig. 11. (a) Secret image. (b) Target image. (c) Camouflage image created from (a) and (b) by Lee et al.'s method. (d) Camouflage image created from (a) and (b) by the proposed method.

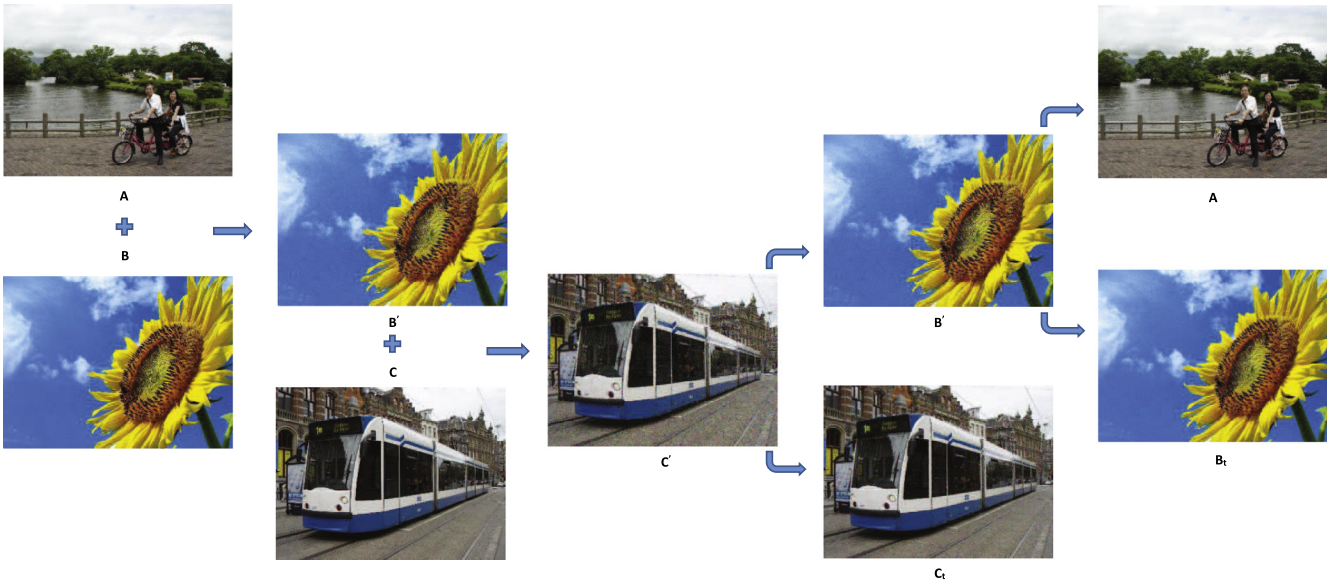


Fig. 12. *A* is a secret image. *B* is a target image. B' is the camouflage image with RMSE = 24.01 with respect to *B*. *C* is a target image. C' is the camouflage image with RMSE = 31.58 with respect to *C*. C_t is the transformed image recovered from C' with RMSE = 22.35 with respect to *C*. B_t is the transformed image recovered from B' with RMSE = 19.84 with respect to *B*.

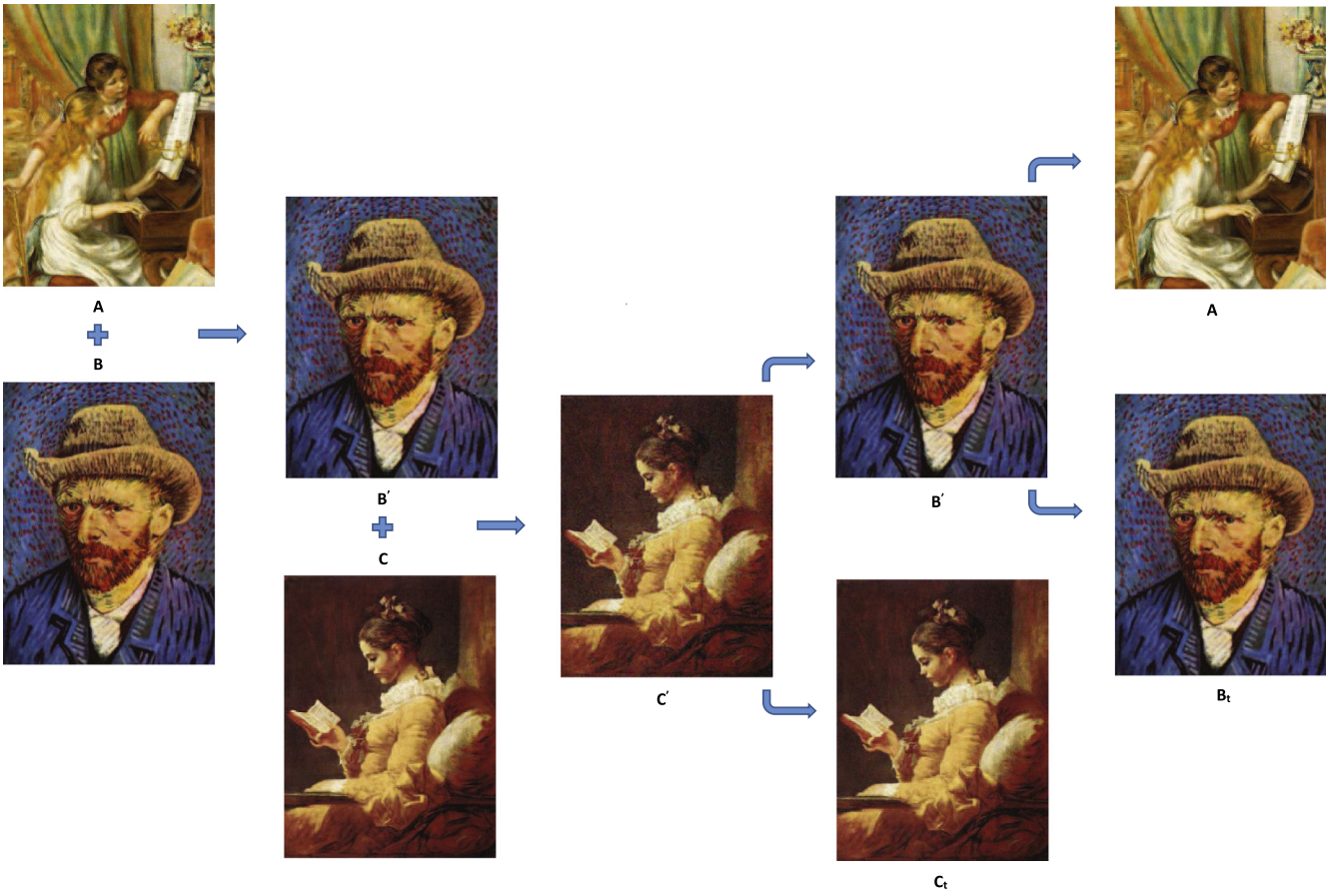
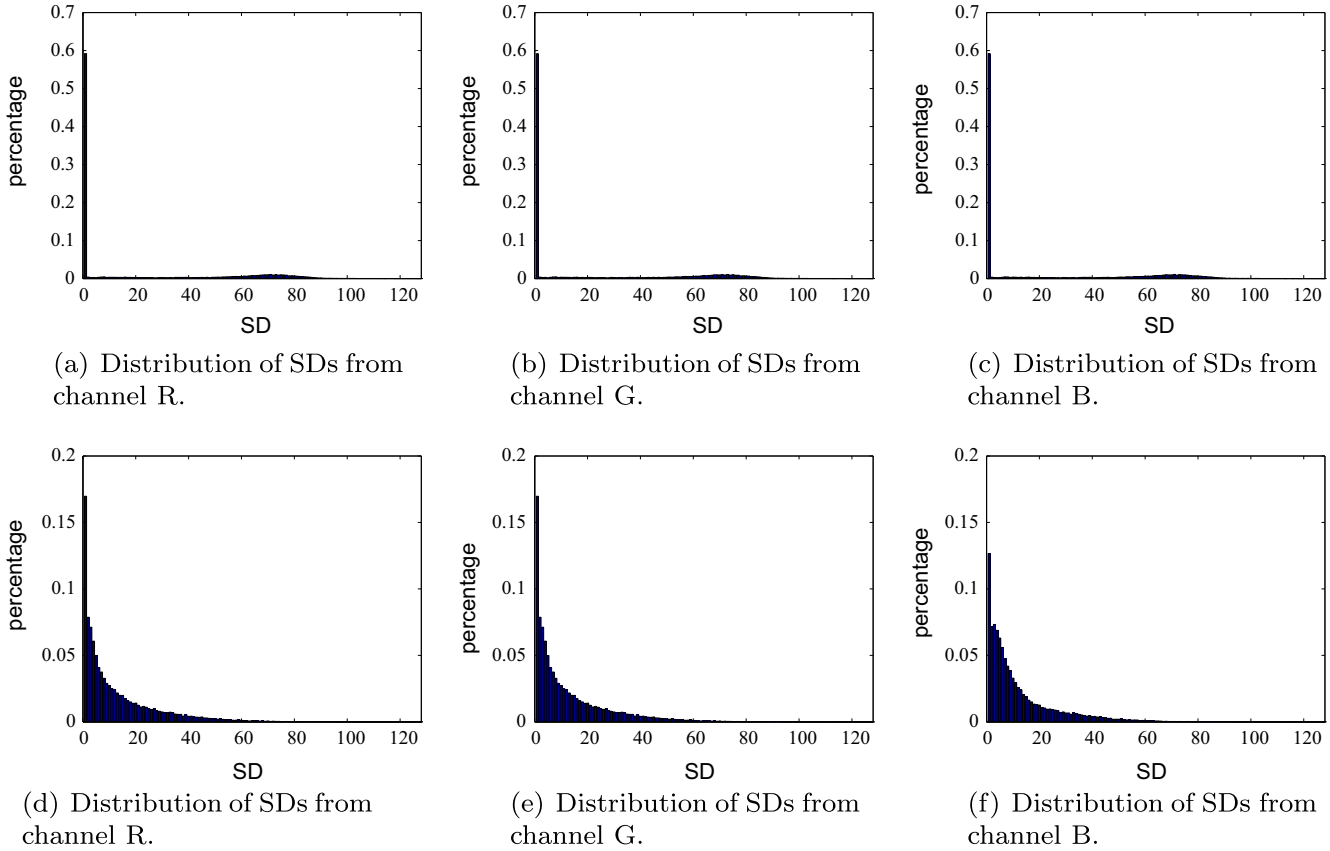


Fig. 13. *A* is a secret image. *B* is a target image. B' is the camouflage image with RMSE = 14.45 with respect to *B*. *C* is a target image. C' is the camouflage image with RMSE = 22.52 with respect to *C*. C_t is the transformed image recovered from C' with RMSE = 14.30 with respect to *C*. B_t is the transformed image recovered from B' with RMSE = 11.35 with respect to *B*.

recovery in the above examples is shown by bpp in Table 5. From Table 5 we can see that, when tile size is set as 8×8 in Lee et al.'s method, the amount of the accessorial information is slightly less than the proposed method for 4×4 tile size. However, if we set the tile size to be 4×4 in Lee et al.'s method, the amount of the accessorial information will be much larger, mostly around 2

(a) Secret image. (b) Target image. (c) Camouflage image.

Fig. 14. (a) Secret image. (b) Target image. (c) Camouflage image created from (a) and (b).



(a) Distribution of SDs from channel R. (b) Distribution of SDs from channel G. (c) Distribution of SDs from channel B.

(d) Distribution of SDs from channel R. (e) Distribution of SDs from channel G. (f) Distribution of SDs from channel B.

Fig. 15. The distributions of SDs of 4×4 tiles from Fig. 14(a) and (b). The above three figures are for image Fig. 14(a), and the below three figures are for image Fig. 14(b).

bpp. So much accessorial information is very difficult to be embedded into an image, especially mosaic image.

To verify the feasibility of the proposed method, we test it on 100 pairs of images, which are randomly depicted from the BossBase image database [22] and sampled down to 1024×1024. The average results of the proposed method and Lee et al.'s method are listed in Table 6.

4.3. Two-round transformation

As mentioned in Section 1, when the image transformation is reversible, it can be used to realize multi-round transformation, i.e., transform the camouflage image to another target image, as shown in Fig. 1(c). By the proposed method we can realize two-round transformation with acceptable image quality.

In 1-round transformation, the sender transforms the secret image A to the target image B and gets a transformed image B_t, then embeds the accessorial information into B_t with an RDH scheme to generate the ultimate camouflage image B'. After getting B', the receiver firstly extracts the accessorial information and restores the transformed image B_t and then restores the secret image A.

In 2-round transformation, the sender firstly transforms A to its target image B getting B_t and then embeds the accessorial information into B_t with an RDH scheme to generate the image B'. Secondly, the sender transforms B' to its target image C to get a transformed image C_t, and then embeds the accessorial information into C_t to generate the ultimate two-layered camouflage image C'. At the receiver side, the receiver firstly extracts the accessorial information from C' and restores C_t and B'. Secondly, the receiver extracts the accessorial information from B' and restores B_t and A. In other words, the receiver gets C_t, B_t, and A.

Two series of experiments on 2-round transformation are shown in Figs. 12 and 13. It can be seen that the ultimate camouflage image C' still has acceptable quality. We also find that, the quality of C' is much worse than that of C_t, while the difference between B' and B_t is not so obvious. In fact, although there are two target images B and C, 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 the image more seriously, which is also the reason that so far the proposed method can only support 2-round transformation.

5. Conclusion and discussion

In this paper, we propose a reversible image transformation method, which can transform a secret image into a freely-selected target image, getting a camouflage image used as the camouflage of secret image with good visual quality, and the secret image can be restored without any loss.

However, in our scheme, the distributions of SDs of secret image and target image are regarded similar. If the distributions of such two images are quite different, the result yielded by the proposed method will be not so good. Take Fig. 14 for example, the distributions of SDs are much different from each other as shown in Fig. 15, so the created camouflage image Fig. 14(c) is poor.

Because of the reversibility, we can realize 2-round transformation, that is, transform the camouflage image to another target image. However, limited by the RDH techniques, the proposed method can only transform images by two rounds in order to keep an acceptable result. How to realize multi-round (e.g., 3 or 4 rounds) transformation is still a challenging problem. Our further work includes improving the transformation and RDH methods to increase the number of transformation rounds.

References

[1] 2014 Celebrity Photo Hack, [Online]. Available: <http://en.wikipedia.org/wiki/2014_celebrity_photo_hack>.
[2] T. Filler, J. Fridrich, Gibbs construction in steganography, IEEE Trans. Inform. Foren. Secur. 5 (4) (2010) 705–720.
[3] V. Holub, J. Fridrich, Designing steganographic distortion using directional filters, in: 2012 IEEE International Workshop on Information Forensics and Security, 2012, pp. 234–239.
[4] V. Holub, J. Fridrich, T. Denemark, Universal distortion function for steganography in an arbitrary domain, EURASIP J. Inform. Secur. 2014 (1) (2014) 1–13.
[5] T. Pevny, P. Bas, J. Fridrich, Steganalysis by subtractive pixel adjacency matrix, IEEE Trans. Inform. Foren. Secur. 5 (2) (2010) 215–224.
[6] J. Kodovskỳ, T. Pevnỳ, J. Fridrich, Modern steganalysis can detect YASS, in: Proc. SPIE, Electronic Imaging, Media Forensics and Security XII, San Jose, CA, 2010.
[7] J. Fridrich, J. Kodovsky, Rich models for steganalysis of digital images, IEEE Trans. Inform. Foren. Secur. 7 (3) (2012) 868–882.
[8] C.-C. Thien, J.-C. Lin, Secret image sharing, Comput. Graph. 26 (5) (2002) 765–770.
[9] C.-N. Yang, J.-F. Ouyang, L. Harn, Steganography and authentication in image sharing without parity bits, Opt. Commun. 285 (7) (2012) 1725–1735.
[10] G. Blakley, Safeguarding cryptographic keys, Proceedings of the National Computer Conference, vol. 48, 1979, pp. 313–317.
[11] A. Shamir, How to share a secret, Commun. ACM 22 (11) (1979) 612–613.
[12] I.-J. Lai, W.-H. Tsai, Secret-fragment-visible mosaic image – a new computer art and its application to information hiding, IEEE Trans. Inform. Foren. Secur. 6 (3) (2011) 936–945.
[13] Y.-L. Lee, W.-H. Tsai, A new secure image transmission technique via secret-fragment-visible mosaic images by nearly reversible color transformations, IEEE Trans. Circ. Syst. Video Technol. 24 (4) (2014) 695–703.
[14] E. Reinhard, M. Ashikhmin, B. Gooch, P. Shirley, Color transfer between images, IEEE Comput. Graph. Appl. 21 (5) (2001) 34–41.
[15] C. Yan, Y. Zhang, J. Xu, F. Dai, L. Li, Q. Dai, F. Wu, A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors, IEEE Signal Process. Lett. 21 (5) (2014) 573–576.
[16] C. Yan, Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai, F. Wu, Efficient parallel framework for HEVC motion estimation on many-core processors, IEEE Trans. Circ. Syst. Video Technol. 24 (12) (2014) 2077–2089.
[17] D. Coltuc, J.-M. Chassery, Very fast watermarking by reversible contrast mapping, IEEE Signal Process. Lett. 14 (4) (2007) 255–258.
[18] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y.Q. Shi, Reversible watermarking algorithm using sorting and prediction, IEEE Trans. Circ. Syst. Video Technol. 19 (7) (2009) 989–999.
[19] W. Zhang, X. Hu, X. Li, N. Yu, Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression, IEEE Trans. Image Process. 22 (7) (2013) 2775–2785.
[20] W.-L. Tai, C.-M. Yeh, C.-C. Chang, Reversible data hiding based on histogram modification of pixel differences, IEEE Trans. Circ. Syst. Video Technol. 19 (6) (2009) 906–910.
[21] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[22] BossBase Image Database, [Online]. Available: <http://agents.fel.cvut.cz/stegodata/RAWs/>.