



Steganography in animated emoji using self-reference

Zhiying Zhu¹ · Qichao Ying¹ · Zhenxing Qian¹ · Xinpeng Zhang¹

Received: 19 July 2020 / Accepted: 19 November 2020 / Published online: 6 January 2021
© The Author(s) 2021

Abstract

Animated emoji is a kind of GIF image, which is widely used in online social networks (OSN) for its efficiency in transmitting vivid and personalized information. Aiming at realizing covert communication in animated emoji, this paper proposes an improved steganography framework in animated emoji. We propose a self-reference algorithm to improve the steganography security. Meanwhile, the relations between adjacent frames of the cover GIF image are considered to further improve the distortion function. After that we embed the secret message into the GIF image using the popular framework of Syndrome Trellis Coding (STC). Experimental results show that the proposed method can provide better security performances than state-of-the-art works.

Keywords Steganography · Animated GIF · Distortion function · Information hiding

1 Introduction

Different from the image formats like JPEG or TIFF, the GIF (Graphics Interchange Format) image is composed of a color palette and a set of index values. For its prevalence in social network applications, it is quite suitable for covert communication by hiding secret data into GIF images. GIF images can be divided into two categories, namely, the static GIF images and the dynamic GIF images. The dynamic is more popular in online social networks (OSN). Animated GIF is a type of dynamic GIF, it is often used to enrich people's social expressions and emotional performance. The most popular GIF is the animated emoji, which is now widely used in OSN like WeChat, Twitter, and Weibo.

Steganography is a method of embedding secret messages into digital covers without introducing serious distortion [1]. In the early times, there were many steganography methods, such as LSB (Least Significant Bits) replacement, F5 [2], etc. LSB replacement is the simplest steganography method [3]. By modifying the least significant binary

of pixel to store information, the human eye cannot perceive the changes. F5 uses matrix embedding to hide secret messages into the JPEG images. However, these methods are fragile against modern steganography analysis. Recently, the Syndrome Trellis Coding (STC) framework is popular for steganography [4], which tries to minimize the additive distortion between the cover and the stego using a predefined distortion function. Generally, the distortion function assigns different distortion costs to different elements of cover. For the spatial images, HILL (High-pass, Low-pass, and Low-pass) [5], WOW (Wavelet Obtained Weights) [6], and SUNIWARD (Spatial UNIversal Wavelet Relative Distortion) [7, 8] are widely proposed. Meanwhile, JUNIWARD (JPEG UNIversal WAvelet Relative Distortion) [7, 8], UED (UNIversal WAvelet Relative Distortion) [9], UERD (Uniform Embedding Revisited Distortion) [10] and HDS (Hybrid Distortion Steganography) [11] are widely used for JPEG images, in which the coefficients of the transfer domain are modified according to the distortion costs. Besides, adaptive methods are proposed to guide the modification direction of the coefficients, which can often improve the security of the modified image [12–15].

As an adversary, steganalysis is used to break steganography, which analyzes the features of an image to determine whether it contains secret messages [16]. The rapid development of steganalysis has brought a great challenge to steganography. Generally, the steganalysis is a kind of classifier that learns the differences of the features between the cover

✉ Zhenxing Qian
zxqian@fudan.edu.cn

✉ Xinpeng Zhang
zhangxinpeng@fudan.edu.cn

¹ Shanghai Institute of Intelligent Electronics and Systems,
School of Computer Science, Fudan University,
Shanghai 200433, China

and the stego [17]. The security of a steganography method can be evaluated by observing the accuracy of the classifier. There are many feature extraction methods, e.g., SPAM (subtractive pixel adjacency model) [18], SRM (Spatial Rich Model) [19], DCTR (Discrete Cosine Transform Residual) [20], and GFR (Gabor Filters Residual) [21].

As the GIF images are widely used, researchers are paying more attention to GIF steganography. In [22], the first steganography algorithm is proposed for indexed images such as static GIF. The scheme searches for the closest color in the palette to reduce the distortion caused by data hiding. In [23], adaptive strategies are proposed to determine which pixels should be modified to embed data. To the best of our knowledge, the first method of embedding data into dynamic GIF images is proposed in [24]. Subsequently, more steganography approaches for animated GIF are proposed [25–28]. In [28], the researchers propose a framework to embed data into animated GIF using the difference between adjacent pixels in the same frame. In [29], a method is proposed to hide data into the animated emoji GIF using the STC framework, in which the distortion functions are improved to achieve better security.

In this paper, we propose a steganography scheme in animated emoji using self-reference, in which we integrate the data embedding impacts for the intra frames and the inter frames. We also provide an algorithm of generating a reference image for guiding the data embedding. With these algorithms, we can achieve a better performance of countering steganalysis. The rest of this paper is organized as follows. We introduce the backgrounds of GIF steganography in Section II. The proposed framework is described in Section III. Section IV shows the experimental results and analysis. Section V concludes the whole paper.

2 Preliminaries

Let there be K frames in an emoji GIF image. Each frame is a color index matrix \mathbf{I} with the size of $M \times N$. The image contains a color palette, in which a limited amount of colors are represented, e.g., 256 colors for an 8-bit palette.

The pixels are represented by I_{ij} , where $i \in \{1, 2, \dots, M\}$ and $j \in \{1, 2, \dots, N\}$. The value of each pixel is represented by an index l defined in the palette C_l , where $l \in \{0, 1, 2, \dots, 255\}$. Accordingly, an RGB value (R_{ij}, G_{ij}, B_{ij}) can be constructed from the index I_{ij} . Figure 1 illustrates the composition of an emoji GIF image.

We denote the cover and the stego images as \mathbf{X} and \mathbf{Y} , respectively. The pixels are represented by X_{ij} and Y_{ij} . After embedding data into any pixel X_{ij} in \mathbf{X} , we obtain the pixel Y_{ij} and the stego \mathbf{Y} . The modification is either binary or ternary. In ternary embedding, each pixel in the stego is $Y_{ij} \in \{X_{ij} + 1, X_{ij}, X_{ij} - 1\}$.

To minimize the change of RGB values caused by data modification, the method in [29] proposes a palette sorting algorithm. First, it calculates the square sum of the pixel values of the RGB channels corresponding to the l -th index value in the palette C_l .

$$t_{(l)} = R(l)^2 + G(l)^2 + B(l)^2 \quad (1)$$

After ascendingly sorting the obtained values, we obtain a sorted palette C'_l . Based on the new palette, we can regenerate a new index matrix I'_k , where k represent the k -th frame.

Let the embedding costs of ternary embedding be ρ_{ij}^+ , ρ_{ij} and ρ_{ij}^- , respectively, where $\rho_{ij} = 0$, $\rho_{ij}^+ \in (0, +\infty)$, $\rho_{ij}^- \in (0, +\infty)$. The generated additive distortion function $D(\mathbf{X}, \mathbf{Y})$ is the sum of the embedding costs of all pixels.

$$D(\mathbf{X}, \mathbf{Y}) = \sum_{i=1, j=1}^{i=M, j=N} \rho_{ij}(X_{ij}, Y_{ij}) \quad (2)$$

To embed a secret message into cover \mathbf{X} , the framework of Syndrome Trellis Coding (STC) requires a modification probability p_{ij} for each pixel. According to [32], the modification probability p_{ij} and embedding cost ρ_{ij} can be calculated by (3).

$$p_{ij}^{(l)} = \frac{e^{-\lambda \rho_{ij}^{(l)}}}{\sum_{l \in \{+1, 0, -1\}} e^{-\lambda \rho_{ij}^{(l)}}} \quad (3)$$

In (3), when the embedding is binary, $|l|=2$; and when it is ternary, $|l|=3$. Because the embedding cost ρ_{ij} is known, p_{ij} can be placed in (4) to obtain the parameter λ , where the m is the amount of secreta data to be embedded by the data-hider.

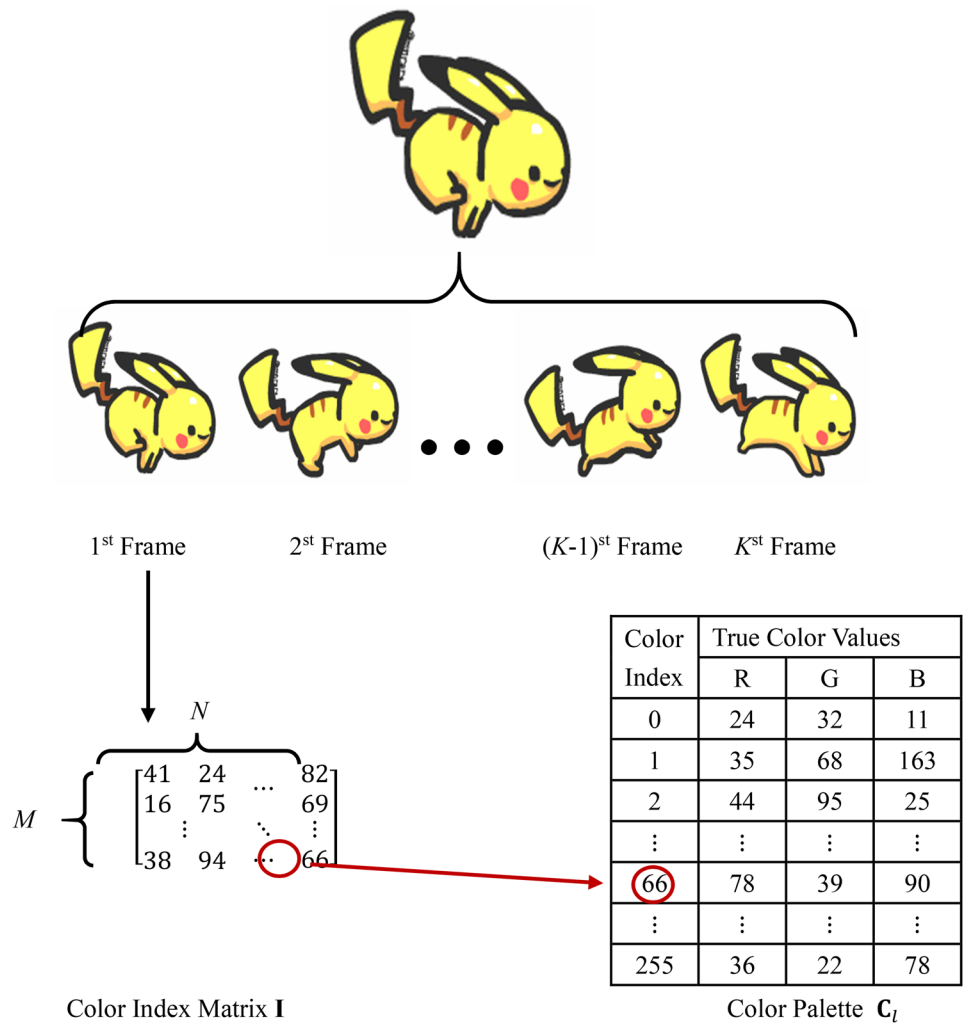
$$H(p) = - \sum_{i=1, j=1}^{i=M, j=N} p_{ij} \log p_{ij} = m \quad (4)$$

3 Proposed framework

The proposed scheme is depicted in Fig. 2. First, after the sort the palette, we decompose the animated GIF into several frames. For each frame, we retrieve the RGB values of every pixel according to the GIF color palette and convert each frame I'_k into a color image F_k , where k is the frame index. Then, we construct a reference frame \hat{F}_k for each frame. With \hat{F}_k we optimize the embedding costs. We further improve the inter-frame distortion using the previous frame as a reference. After embedding data into each frame, we obtain a stego GIF.

A. Improved bipolar embedding.

Fig. 1 Illustration of the composition of an emoji GIF image



Since GIF is a compressed format, it can be regarded as a 256-color image compressed from a true-color image. Therefore, we can use the original content of the image before GIF compression to improve Bipolar Embedding. We convert the GIF frames into the color images $F = \{F_1, \dots, F_K\}$ according to the palette, where K is the number of frames. For each frame, every pixel $B = (R_{ij}, G_{ij}, B_{ij})$ has a corresponding pixel $A = (\hat{R}_{ij}, \hat{G}_{ij}, \hat{B}_{ij})$ at the same location before GIF compression. It is equivalent to shift the RGB value from point A to point B . Vector AB stands for the distortion during compression.

When we embed secret messages into I_{ij} , the pixels are either added or subtracted by one. In (5), a refers to the Hamming distance between A and B . We also use $C = (R_{ij}^+, G_{ij}^+, B_{ij}^+)$ or $(R_{ij}^-, G_{ij}^-, B_{ij}^-)$ as the corresponding pixel at the same location after embedding. In (6) and (7), the Hamming distances caused by + 1 and - 1 operation are defined as b^+ and b^- , respectively.

$$a = [(R_{ij} - \hat{R}_{ij}), (G_{ij} - \hat{G}_{ij}), (B_{ij} - \hat{B}_{ij})] \tag{5}$$

$$b^+ = [(R_{ij}^+ - R_{ij}), (G_{ij}^+ - G_{ij}), (B_{ij}^+ - B_{ij})] \tag{6}$$

$$b^- = [(R_{ij}^- - R_{ij}), (G_{ij}^- - G_{ij}), (B_{ij}^- - B_{ij})] \tag{7}$$

Subsequently, we define the modification angle between a and b^+ or b^- as θ^+ or θ^- in (8) and (9). The operator $|\cdot|$ stands for the module of vector.

$$\theta^+ = \arccos \left(\frac{a \cdot b^+}{|a| \cdot |b^+|} \right) \tag{8}$$

$$\theta^- = \arccos \left(\frac{a \cdot b^-}{|a| \cdot |b^-|} \right) \tag{9}$$

Fig. 2 Overview of the proposed scheme

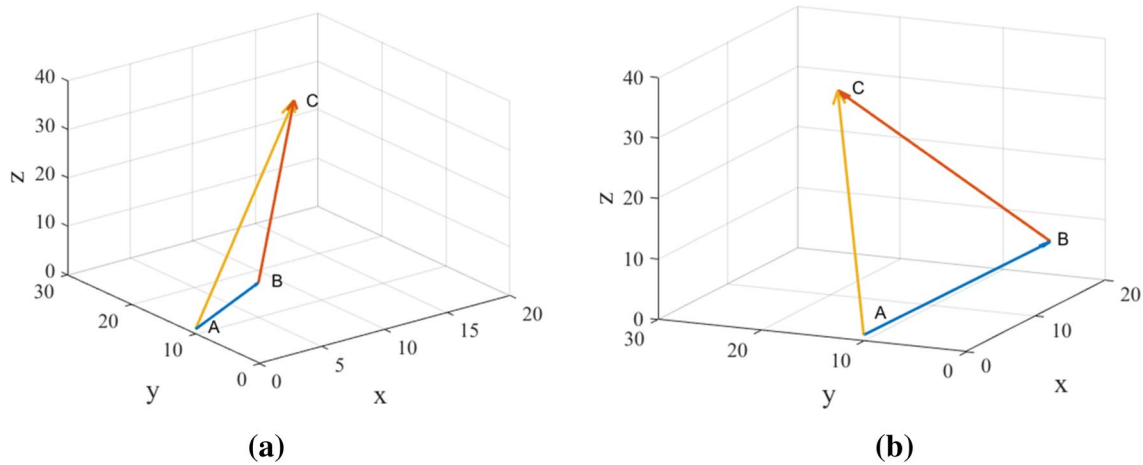
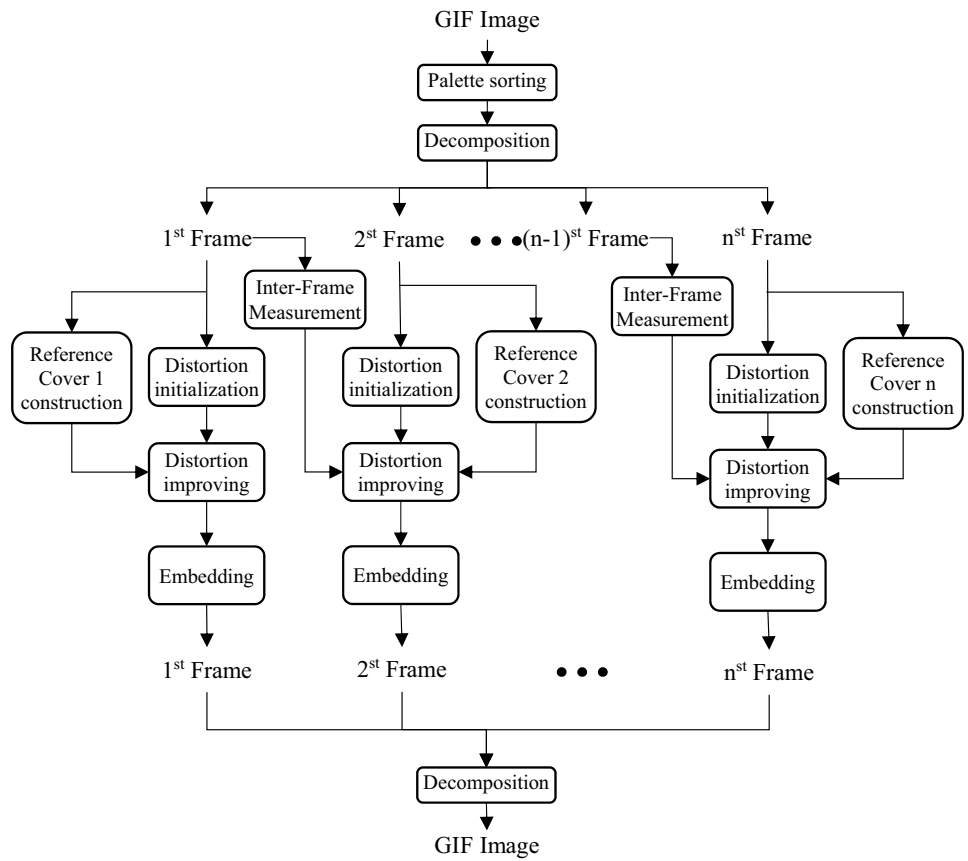


Fig. 3 Pixel modification for GIF where modification angle is: **a** acute, **b** obtuse

Figure 3 illustrates the cases that the modification angles are acute or obtuse, respectively. Vector AB stands for the distortion during compression and vector BC stands for the distortion caused by data embedding.

In Fig. 3a, when θ between the two vectors AB and BC is acute, the compression direction is the same as the embedding direction. In the other words, the extra error

would be the embedding error “plus” the compression error. In Fig. 3b, when θ is an obtuse angle, the compression direction and embedding direction are the opposite. Then, the extra error would be the embedding error “minus” the compression error. We denote the extra error AC in Fig. 3 as c^+ and c^- , in (10).

$$\begin{cases} c^+ = a + b^+ \\ c^- = a + b^- \end{cases} \quad (10)$$

In summary, when $\theta^+ \in (0, \pi/2)$ or $\theta^- \in (0, \pi/2)$, $|c^+|$ or $|c^-|$ is larger than each element in $\{|a|, |b^+|, |b^-|\}$. When $\theta^+ \in (\pi/2, \pi)$ or $\theta^- \in (\pi/2, \pi)$, c^+ or c^- is smaller than any one in $\{|a|, |b^+|, |b^-|\}$. Therefore, when θ^+ or θ^- is obtuse, $|c|$ would be smaller. Therefore, we prefer the pixel modification when the angle θ^+ or θ^- be obtuse, and restrict data embedding when θ^+ or θ^- is acute.

This algorithm reduces the extra error caused by embedding modification and makes the compressed image closer to the original image. Thus, it can improve the security of steganography effectively.

B. Reference construction.

According to the aforementioned analysis, if a data-hider have the original content of the image before GIF compression, a better performance of a security can be achieved by modifying the pixel values toward the original values. However, in most cases, the data hider does not have the original content of the images before compression. Therefore, we use an algorithm to construct a reference image for each frame.

To achieve a satisfactory performance, the constructed reference images should be close to the original image before GIF compression. Inspired by [30], we can treat the image compression as a procedure of adding noise into the original content. To remove this kind of noise, we propose to use the DnCNN model proposed in [31] to construct a reference image. This model has been proved to be useful in many denoising tasks. Different from the existing denoising methods that are defined for additive white Gaussian noise

at a certain noise level, the DnCNN model is able to handle Gaussian denoising with the unknown noise level. Besides, this model is able to handle multiple general image denoising tasks, such as Gaussian denoising, single image super-resolution, and JPEG image deblocking.

Denote the luminance of the original image as Y_{Ori} and the GIF-compressed image as Y_{Comp} . As shown in Fig. 4, the residual image is the difference between the original image and compressed image in the luminance channel. We denote Y_{Res} as the residual image, where $Y_{Ori} = Y_{Comp} + Y_{Res}$. In other words, the residual image can be regarded as a kind of image noise. With a residual learning strategy, the residual image can be estimated [31]. DnCNN is trained on the luminance channel because human perception is more sensitive to changes in brightness than changes in chrominance.

In Fig. 5, an animated emoji is decomposed into several frames, then the GIF-compressed images are converted from RGB space to the $YCbCr$ space. The DnCNN network is trained to detect the residual images from the luminance of the color frames. Three different colors represent three types of layers. For the first layer, marked in yellow, 64 filters are used to generate 64 feature maps. ReLU is nonlinearity activation function. For layers 2~($D - 1$), marked in blue, 64 filters sized $3 \times 3 \times 64$ are used. The batch normalization is added between convolution and ReLU. It is incorporated to speed up training as well as boost the denoising performance. The orange layer is the last layer, in which filters sized $3 \times 3 \times 64$ are used to reconstruct the output. D is the depth of DnCNN. For an image denoising task, the depth of network (the number of convolution layers) is generally specified as 20.

With the model of DnCNN, we can reconstruct an undistorted version of a compressed frame by subtracting the

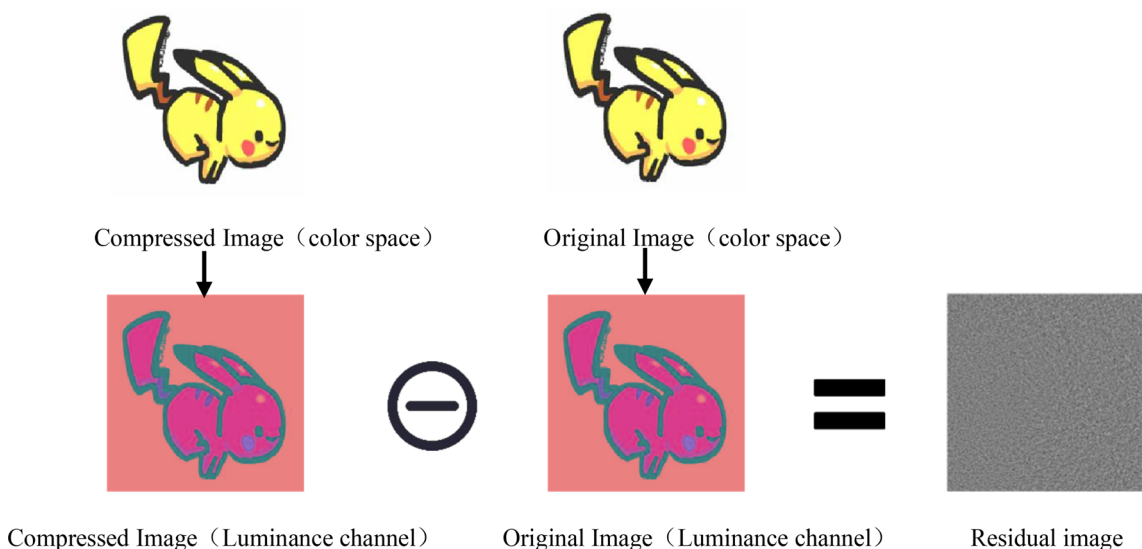


Fig. 4 The generation of residual image

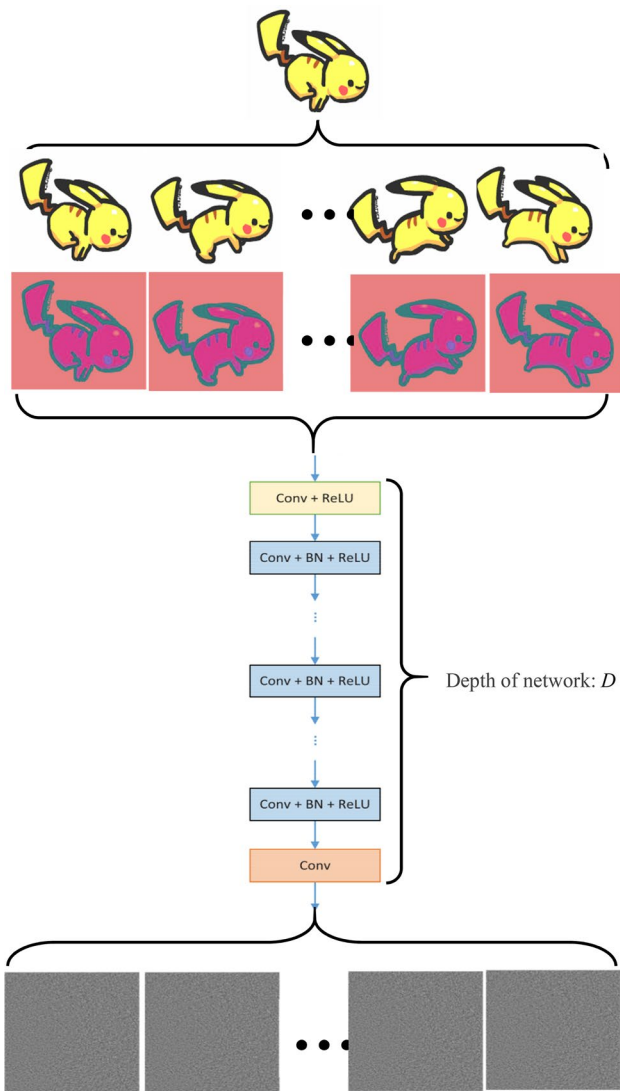


Fig. 5 The architecture of residual image construction network

compressed luminance channel from a residual image, and then converts the image back to the RGB color space.

We denote the k -th RGB frame of GIF as F_k , which is constructed from I'_k and C'_l . Let the reference frame be \hat{F}_k . $Y_{\hat{F}_k}$ and Y_{F_k} are the luminance channel of \hat{F}_k and F_k . The reference frame $Y_{\hat{F}_k}$ can be calculated by (11)

$$Y_{\hat{F}_k} = Y_{F_k} - \text{Dn_CNN}(Y_{F_k}) \tag{11}$$

where $\text{Dn_CNN}(\cdot)$ represents the DnCNN denoising network. Concatenate the denoised luminance channel $Y_{\hat{F}_k}$ with the original chrominance channels to obtain the denoised image in the $YCbCr$ space. We further convert the $YCbCr$ image to RGB to generate the reference image \hat{F}_k . The reference image \hat{F}_k is close to the original image \tilde{F}_k .

C. Distortion function improvement.

Let ρ_{ij}^+ and ρ_{ij}^- be the embedding cost for +1 and -1 in the intra-frame embedding, respectively, where $i \in \{1, \dots, M\}, j \in \{1, \dots, N\}$. In many steganography methods based on STC, ρ_{ij}^+ is identical to ρ_{ij}^- . In the proposed method, we improve the embedding cost function according to the RGB value $(\hat{R}_{ij}, \hat{G}_{ij}, \hat{B}_{ij})$ of the reference image \hat{F} .

We first initialize the original costs ρ_{ij} for the pixels in each frame using the traditional distortion functions like HILL [5], WOW [6] and UNIWARD [7, 8]. For each pixel, we multiply the original cost ρ_{ij} by a factor α . There are two cases for the factor α when performing ± 1 operations, namely, α^+ or α^- , which are depicted in (12) and (13). We adjust the distortion function in (14) and (15) by combining three optimization factors, in which wetCost represents a very large value, e.g., 10^8 in the experiments.

$$\alpha^+ = |a + b^+| / (|a| + |b^+|) \tag{12}$$

$$\alpha^- = |a + b^-| / (|a| + |b^-|) \tag{13}$$

$$\rho_{ij}^+ = \begin{cases} \text{wetCost} & \text{if } \theta^+ \in (0, \pi/2) \\ \rho_{ij} & \text{if } \theta^+ = \pi/2 \\ \alpha^+ \cdot \rho_{ij} & \text{if } \theta^+ \in (\pi/2, \pi) \end{cases} \tag{14}$$

$$\rho_{ij}^- = \begin{cases} \text{wetCost} & \text{if } \theta^- \in (0, \pi/2) \\ \rho_{ij} & \text{if } \theta^- = \pi/2 \\ \alpha^- \cdot \rho_{ij} & \text{if } \theta^- \in (\pi/2, \pi) \end{cases} \tag{15}$$

On the other hand, during data hiding, there would be differences between adjacent frames. Therefore, we must consider the impact of inter-frame embedding. For each frame F_k , we use the previous frame F_{k-1} as reference. The RGB values $(R_{ij}^{k-1}, G_{ij}^{k-1}, B_{ij}^{k-1})$ from F_{k-1} is used to guide the modification of the current frame. The procedures of inter-frame embedding are similar except that a is redefined in (16) and the cost for inter-frame embedding are redefined as ρ_{ij}^+ and ρ_{ij}^- in (17) and (18). Unlike a defined in (5), a redefined in (16) represents the change of direction of RGB values at the same position of adjacent frames. Then we get α^+ and α^- by applying (12) and (13), and update the distortion function as (17) and (18).

$$a = \left[\left(R_{ij} - R_{ij}^{k-1} \right), \left(G_{ij} - G_{ij}^{k-1} \right), \left(B_{ij} - B_{ij}^{k-1} \right) \right] \tag{16}$$

$$\rho_{ij}^+ = \begin{cases} \text{wetCost} & \text{if } \theta^+ \in (0, \pi/2) \\ \rho_{ij} & \text{if } \theta^+ = \pi/2 \\ \alpha^+ \cdot \rho_{ij} & \text{if } \theta^+ \in (\pi/2, \pi) \end{cases} \quad (17)$$

$$\rho_{ij}^- = \begin{cases} \text{wetCost} & \text{if } \theta^- \in (0, \pi/2) \\ \rho_{ij} & \text{if } \theta^- = \pi/2 \\ \alpha^- \cdot \rho_{ij} & \text{if } \theta^- \in (\pi/2, \pi) \end{cases} \quad (18)$$

Finally, we combine the cost in intra-frame and inter-frame embedding, and obtain the final distortion function in (19).

$$\begin{cases} \bar{\rho}_{ij}^+ = \rho_{ij}^+ \times \rho_{ij}^+ \\ \bar{\rho}_{ij}^- = \rho_{ij}^- \times \rho_{ij}^- \end{cases} \quad (19)$$

D. Payload allocation.

For most animated GIF, the patterns on the different frame are different. To improve data security of each frame, we adaptively allocate different payloads to the frames according to their characteristics. We adopt the algorithm proposed in [32] for the purpose, in which an m -bit secret message is embedded into n covers with a minimized distortion. The distortion is calculated in (20),

$$D_{\min}(m, n, \rho) = \sum_{i=1, j=1}^{i=M, j=N} \rho_{ij} p_{ij} \quad (20)$$

and the optimization problem is defined in (21),

$$\begin{aligned} \min_{D_{\min}} D_{\min}(m, n, \rho) \\ \text{subject to } H(p) = m \end{aligned} \quad (21)$$

In this paper, the optimization problem is defined as (22),

$$\begin{aligned} \min_{D_{\min}} D_{\min}(m, n, \bar{\rho}) = \sum_{k=1}^n \bar{\rho}_k \bar{p}_k \\ \text{subject to } \sum_{k=1}^K H(\bar{p}_k) = m \end{aligned} \quad (22)$$

where n is the sum of the number of all selected GIF frames, $\bar{\rho}_k$ is the embedding cost of the k -th frame, and \bar{p}_k is the embedding possibility of the k -th frame. After calculating the distortion function, we input the embedding costs and the payloads of all frames into the constraints in (22), and obtain the modification probability of each frame. The embedding payload of each frame can be calculated by (23)

$$m_k = \sum_{k=1}^n H(\bar{p}_k) \quad (23)$$

4 Experimental results

To verify the proposed framework, we have conducted many experiments on the emoji GIF dataset provided by [29] that contains 560 animated GIFs. Several examples are shown in Table 1. These GIF images are in 8-bit palette format where each image contains 256 colors.

We use binary pseudo-random sequences as the hidden data, i.e., the possibilities for zero and one are identical. We use the popular HILL, UNIWARD and WOW as initial distortion functions. The reference images are generated by DnCNN. We name the proposed steganography based on the improved versions of HILL, UNIWARD and WOW as PD-HILL, PD-UNIWARD and PD-WOW, respectively.

The embedding tasks are done by the STC framework. The capacity of secret data embedded in each frame are set as 600, 700, 800, 900, 1000, and 1100 bits, respectively. Subsequently, we also use the payloads of 0.05bpp, 0.1bpp, 0.15bpp, 0.2bpp and 0.25bpp for further comparisons.

For steganalysis, we use the ensemble classifier and the feature sets of SPAM and SRMQ1. Half of the cover and stego are used for training and the others are for testing data. The minimal total error P_E is used as the criterion to evaluate the performances of steganography. In (24), P_{FA} is the false alarm rate and P_{MD} is the missed detection rate. The average P_E by 10 random tests is used to evaluate the performance [17].

$$P_E = \min_{P_{FA}} \left(\frac{P_{FA} + P_{MD}}{2} \right) \quad (24)$$

When testing the security of intro-frame steganography, we convert every fame of GIF into a colorful image and transform them into gray images. SPAM or SRMQ1

Table 1 Several examples of the dataset



are used to extract the features. When testing the security of inter-frame steganography, we calculate the difference between two frames, which is used for the subsequent steganalysis.

Table 2 provides the embedding test for an emoji image with different payloads and algorithms. The original HILL method is not effective in embedding a large payload in GIF due to texture simplicity, as obvious pepper and salt noises can be found in the smooth areas. While embedding data using the method in [29], the pepper and salt noises appear on the edges of the image. With our method, there are no obvious noises in the edge and texture areas.

To show the effectiveness of the proposed framework, we use the same experiment setting as [29]. The proposed method PD-HILL, PD-WOW and PD-UNIWARD are used to embed the same amount of message into the same dataset. Table 3 show the testing errors of the PD-HILL, [29] and HILL against SPAM and SRMQ1. The results show that the proposed method has better visual quality as well as security.

We further apply larger embedding payloads, i.e., 0.05 *bpp* ~ 0.25 *bpp*. Many GIF images cannot accommodate large amounts of secret messages when using HILL. Therefore, we only compare our method with [29]. In Fig. 6, we

use different initial distortion functions. The results show that the proposed method outperforms [29] in most cases.

Finally, we conduct the inter-frame security experiments, which are compared with [29]. We use ensemble classifier to calculate the P_E between frames. Table 4 shows the inter-frame testing errors of the PD-HILL and [29]. It can be seen that the proposed method has achieved better performance.

5 Conclusions

In this paper, we propose an improved steganography method for animated emoji using self-reference. We first construct the reference images by DnCNN network. Guided by the reference images, we adaptively modify the pixels according to the Hamming distances in RGB color space after conducting + 1 and - 1 operations. We further use the current frame as a reference to improve the security of steganography between frames. Several typical loss functions such as HILL are used and the embedding is done by the STC framework. Experimental results show that the security performances of the proposed method outperform state-of-the-art steganography method for animated emoji images.

Table 2 Embedding test for an animated GIF under different payloads and algorithms








Original Image	Stego Images			
	method payload	HILL	Shi et al. [29]	Proposed Framework
	0.05 <i>bpp</i>			
	0.25 <i>bpp</i>			

Table 3 Testing errors of the PD-HILL, [29] and HILL against SPAM and SRMQ1 under low capacity

Steganography algorithm	Feature	Capacity (bits)					
		600	700	800	900	1000	1100
PD-HILL	SPAM	0.4736	0.4736	0.4697	0.4687	0.4672	0.4651
	SRMQ1	0.3125	0.2982	0.2896	0.2818	0.2703	0.2639
[29]-Hill	SPAM	0.3085	0.3089	0.3087	0.3091	0.3081	0.3083
	SRMQ1	0.0804	0.0799	0.0785	0.0764	0.0763	0.0762
Hill	SPAM	0.1199	0.1027	0.0879	0.0746	0.0644	0.0574
	SRMQ1	0.0624	0.0540	0.0453	0.0415	0.0368	0.0329

Bold part reflects the advanced nature of the algorithm proposed

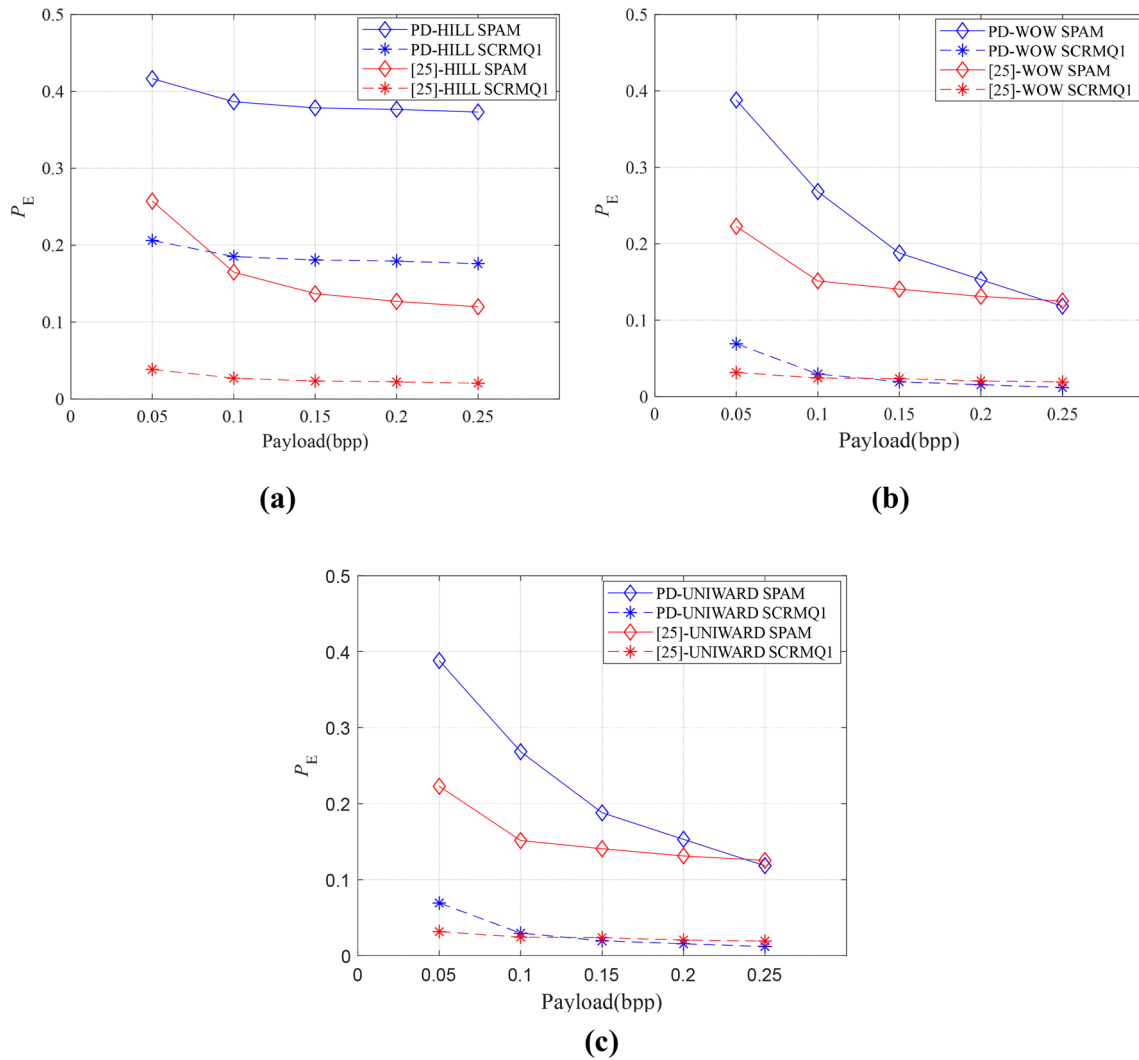


Fig. 6 Comparisons using a HILL b WOW and c UNIWARD

Table 4 The testing errors on inter-frame of the PD-HILL and [29] against SPAM and SRMQ1

Steganography algorithm	Feature	Capacity (bits)				
		0.05	0.1	0.15	0.2	0.25
PD-HILL	SPAM	0.0746	0.0711	0.0675	0.0681	0.0680
	SRMQ1	0.0767	0.0750	0.0751	0.0771	0.0763
[29]-HILL	SPAM	0.0305	0.0260	0.0223	0.0200	0.0173
	SRMQ1	0.0289	0.0264	0.0257	0.0258	0.0277

Bold part reflects the advanced nature of the algorithm proposed

Acknowledgements This research was supported by National Science Foundation of China (Grant U20B205).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes

were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Wang, Z., Zhang, X., Yin, Z.: Joint cover-selection and payload-allocation by steganographic distortion optimization. *IEEE Signal Process. Lett.* **25**(10), 1530–1534 (2018)
2. Westfeld, A.: F5-A steganographic algorithm-high capacity despite better steganalysis. In: *Proceedings of 4th International Workshop on Information hiding*, Pittsburgh, PA, USA, pp. 289–302 (2001)
3. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Information hiding—a survey. *Proc. IEEE* **87**(7), 1062–1078 (1999)
4. Filler, T., Judas, J., Fridrich, J.: Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Trans. Inf. Forensics Secur.* **6**(3), 920–935 (2011)
5. Li, B., Wang, M., Huang, J., Li, X.: A new cost function for spatial image steganography. In: *IEEE International Conference on Image Processing (ICIP)*, Paris, pp. 4206–4210 (2014)
6. Holub, V., Fridrich, J.: Designing steganographic distortion using directional filters. In: *IEEE International Workshop on Information Forensics and Security (WIFS)*, Tenerife, pp. 234–239 (2012)
7. Holub, V., Fridrich, J.: Digital image steganography using universal distortion. In *Proceedings of the first ACM workshop on Information hiding and multimedia security*, New York, NY, USA pp. 59–68 (2013)
8. Holub, V., Fridrich, J., Denemark, T.: ‘Universal distortion function for steganography in an arbitrary domain.’ *EURASIP J. Inf. Sec.* **2014**(1), 1–13 (2014)
9. Guo, L., Ni, J., Shi, Y.Q.: Uniform embedding for efficient JPEG steganography. *IEEE Trans. Inf. Forensics Sec.* **9**(5), 814–825 (2014)
10. Guo, L., Ni, J., Su, W., Tang, C., Shi, Y.: Using statistical image model for JPEG steganography: uniform embedding revisited. *IEEE Trans. Inf. Forensics Sec.* **10**(12), 2669–2680 (2015)
11. Wei, Q., Yin, Z., Wang, Z., Zhang, X.: Distortion function based on residual blocks for JPEG steganography. *Multimed. Tools Appl.* **14**, 17875–17888 (2018)
12. Denemark, T., Fridrich, J.: Side-informed steganography with additive distortion. In: *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, Rome, pp. 1–6 (2015)
13. Denemark, T., Fridrich, J.: Model based steganography with pre-cover. *Electron. Imaging* **2017**(7), 56–66 (2017)
14. Denemark, T., Fridrich, J.: Steganography with multiple JPEG images of the same scene. *IEEE Trans. Inf. Forensics Sec.* **12**(10), 2308–2319 (2017)
15. Wang, Z., Qian, Z., Zhang, X., Yang, M., Ye, D.: On improving distortion functions for JPEG steganography. *IEEE Access* **6**, 74917–74930 (2018)
16. Li, F., Wu, K., Zhang, X., Yu, J., Lei, J., Wen, M.: Robust batch steganography in social networks with non-uniform payload and data decomposition. *IEEE Access* **6**, 29912–29925 (2018)
17. Kodovsky, J., Fridrich, J., Holub, V.: Ensemble classifiers for steganalysis of digital media. *IEEE Trans. Inf. Forensics Sec.* **7**(2), 432–444 (2012)
18. Pevny, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. Inf. Forensics Sec.* **5**(2), 215–224 (2010)
19. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Sec.* **7**(3), 868–882 (2012)
20. Holub, V., Fridrich, J.: Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Trans. Inf. Forensics Sec.* **10**(2), 219–228 (2015)
21. Song, X., Liu, F., Yang, C., Luo, X., Zhang, Y.: Steganalysis of adaptive JPEG steganography using 2D Gabor filters. In: *Proceedings of the 3rd ACM workshop on information hiding and multimedia security*, pp.15–23 (2015)
22. Samarutunge, S.G.K.D.N.: New steganography technique for palette based images. In: *2007 International Conference on Industrial and Information Systems*, Penadeniya, Sri Lanka, pp. 335–339 (2007)
23. Fathurohman, I.T., Purboyo, T.W., Nugrahaeni, R.A.: Comparative analysis of steganography using LSB and adaptive method on GIF image. *Int. J. Appl. Eng. Res.* **12**(21), 10999–11006 (2017)
24. Munir R.: Chaos-based modified “EzStego” algorithm for improving security of message hiding in GIF image. In: *2015 International Conference on Computer, Control, Informatics and Its Applications*, Bandung, Indonesia, pp. 80–84 (2015)
25. Munir, R.: Application of the modified EzStego algorithm for hiding secret messages in the animated GIF images. In: *2016 2nd International Conference on Science in Information Technology*, Balikpapan, Indonesia, pp. 58–62 (2016)
26. Juzar, M.T., Munir, R.: Message hiding in animated GIF using multibit assignment method. In: *2016 International Symposium on Electronics and Smart Devices*, Bandung, Indonesia, pp. 225229 (2016)
27. Munir, R.: Visual cryptography of animated GIF image based on XOR operation. In: *2017 International Conference on Advanced Computing and Applications*, Ho Chi Minh City, Vietnam, pp. 117–121 (2017)
28. Ratan, K.B., Kousik, D., Paramartha, D.: Steganography in grey scale animated GIF using hash based pixel value differencing. In: *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, Kolkata, India, pp. 248–252 (2018)
29. Shi, L., Wang, Z., Qian, Z., et al.: Distortion function for emoji image steganography. *Comput. Mater. Contin.* **58**, 943–953 (2019)
30. Kutter, M., Winkler, S.: A vision-based masking model for spread-spectrum image watermarking. *IEEE Trans. Image Process.* **11**(1), 16–25 (2002)
31. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017)
32. Filler, T., Fridrich, J.: Gibbs construction in steganography. *IEEE Trans. Inf. Forensics Sec.* **5**(4), 705–720 (2010)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.