Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8812087, 13 pages
https://doi.org/10.1155/2020/8812087

Research Article

A New Steganography Method for Dynamic GIF Images Based on Palette Sort

Jingzhi Lin,[1] **Zhenxing Qian**,[2,3] **Zichi Wang**,[1] **Xinpeng Zhang**,[2,3] **and Guorui Feng**[1]

[1]*Shanghai Institute for Advanced Communication and Data Science, School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China*
[2]*Shanghai Institute of Intelligent Electronics and Systems, School of Computer Science and Technology, Fudan University, 201203 Shanghai, China*
[3]*China and Shanghai Institute of Intelligent Electronics & Systems, School of Computer Science, Fudan University, Shanghai 200433, China*

Correspondence should be addressed to Zhenxing Qian; zxqian@fudan.edu.cn

Received 1 June 2020; Revised 29 July 2020; Accepted 5 August 2020; Published 28 August 2020

Academic Editor: Zhili Zhou

This paper proposes a new steganography method for hiding data into dynamic GIF (Graphics Interchange Format) images. When using the STC framework, we propose a new algorithm of cost assignment according to the characteristics of dynamic GIF images, including the image palette and the correlation of interframes. We also propose a payload allocation algorithm for different frames. First, we reorder the palette of GIF images to reduce the modifications on pixel values when modifying the index values. As the different modifications on index values would result in different impacts on pixel values, we assign the elements with less impact on pixel values with small embedding costs. Meanwhile, small embedding costs are also assigned for the elements in the regions that the interframe changes are large enough. Finally, we calculate an appropriate payload for each frame using the embedding probability obtained from the proposed distortion function. Experimental results show that the proposed method has a better security performance than state-of-the-art works.

1. Introduction

Stenography is technology that hides secret data into covers for covert transmission [1]. The most important aim of steganography is to combat the adversary's detection using steganalysis tools [2–4]. As digital images are the most popular covers in steganography, many methods and tools of digital image steganography have been developed since 1990s, such as LSB (Least Significant Bitplane) [5]. To combat early steganalysis attacks, some algorithms have also been designed to keep the invariance of statistical properties [6]. However, most of the traditional works are not secure enough to modern steganalysis.

Currently, content adaptive steganography approaches are designed to improve security by minimizing the distortion between the cover and the stego. According to this idea, a popular framework for digital image steganography is defined in [7, 8], which includes two points, i.e., allocating the embedding cost and realizing the data encoding. In the first part, the embedding cost is allocated for each element of the cover to quantify the effect on the cover image when modifying the elements [9, 10]. In the second part, an encoding method is designed to achieve the theoretical embedding payload with the distortion function. Nowadays, STC (Syndrome-Trellis Codes) is the most popular tool for adaptive embedding [11, 12]. In this framework, defining a suitable distortion function is vital for steganography security. Starting from HUGO [13], many distortion functions have been proposed for spatial and JPEG images, e.g., MVGG [7], WOW [14], UNIWARD [15], HILL [16], and UED [17]. For spatial images, distortion functions allocate the embedding costs for gray values or RGB values. For JPEG images, distortion functions allocate the embedding costs for JPEG coefficients. Besides, some spatial image distortion functions can be used for JPEG images by allocating the embedding costs for the coefficients obtained from the inverse

transformation of JPEG coefficients. As a countering technique, many effective steganalysis approaches have also been proposed. Steganalysts always train models to distinguish suspicious images from clean images [18]. Generally, a steganalysis model contains two parts. One is to extract features from a set of images, such as SRM [18], SPAM [19], DCTR [20], and GFR [21]. The other is to train a classifier using machine learning tools, e.g., the ensemble classifier [22]. Besides, both parts can also be realized by deep learning.

With the development of 4G/5G communication techniques and the social networks, GIF (Graphics Interchange Format) images are more and more popular because of the characteristics of the small size and the convenience to display animation effects. Different from the spatial images and JPEG images, each image frame used in GIF is composed of a color list, i.e., palette, and a set of index values, i.e., image data. When using GIF images as covers, there are three ways to hide secret data. The first way is structure steganography that modifies certain sections in the file header to accommodate secret data. The second way is to embed secret data by changing the correspondence between index values and actual pixel values in the palette, e.g., Gifshuffile [23]. The third way is to embed secret data by slightly modifying the index values, such as OPA (Optimal Parity Assignment) [24] and MBA (MultiBit Assignment) [25].

In 1999, Fridrich proposes a steganography method for palette images, which searches for closest colors and embeds data according to the parity of pixel values [26]. Some other methods for GIF steganography have also proposed in [27–29]. In recent years, there have also been some works on GIF steganography. Generally, these methods use some mathematical operations to calculate whether a pixel should be modified to embed data, e.g., the adaptive method using MOD operation by Fathurohman et al. [30]. These works have good capabilities of data embedding. However, they were designed for static GIF images.

In 2015, a method for embedding data into dynamic GIF images is proposed, which uses EzStego and chaos system to hide data [31]. From 2016 to 2017, a series of steganography approaches for dynamic GIF images were proposed [32–34]. In those methods, the steganography for static GIF is directly used in each frame of the dynamic GIF. Meanwhile, each frame is assigned with an equal payload. In 2018, Basak et al. propose to embed data into dynamic GIF images using the difference between adjacent pixels in the same frame [35]. Saleh and Merzah propose to combine LZW, EzStego, and LSB to embed secret text messages into dynamic GIF images [36]. Shi et al. propose to use the STC framework for emoji GIF images by designing a new distortion function [37]. These works have good performances on dynamic GIF steganography. However, the correlations between frames are not investigated efficiently. To design a more secure steganography tool for dynamic GIF images, we must consider the changes in the interframes. Meanwhile, payload assignment is also important [38]. While previous works embed even payloads into GIF frames, it would be more appropriate to distribute payloads unevenly to different frames to improve the performance of statistical undetectability, since each frame has different contents.

In this paper, we propose a new algorithm of cost assignment for dynamic GIF images by combining the characteristics of dynamic GIF image palette and the inter frames. Meanwhile, we propose to assign suitable embedding payloads for each frame using the entropy function. Finally, we construct a steganography framework for dynamic GIF images to improve the security. The rest of this paper is organized as follows. We introduce the preliminaries of GIF steganography in Section 2. The proposed framework will be described in Section 3. Section 4 shows the experimental results and analysis. Section 5 concludes the whole paper.

2. Preliminary

In this paper, the matrices, vectors, and sets are written in boldface, the variables are written in italics. For a dynamic GIF image $\mathbf{A}$ with t frames, we denote the index values of the dynamic GIF image as $\mathbf{a}$. For each frame, we denote the jth frame as $\mathbf{A}_j$ and denote the index value sequence of $\mathbf{A}_j$ as $\mathbf{a}_j = \{a_{1j}, a_{2j}, \cdots, a_{nj}\}$. Therefore, $\mathbf{a}$ can represented as $\{\{a_{11}, a_{21}, \cdots, a_{n1}\}, \cdots, \{a_{1t}, a_{2t}, \cdots, a_{nt}\}\}$, in which n is the number of pixels in a frame and t is the number of frames in the dynamic GIF image. For a color dynamic GIF image, the pixel A_{ij} of the ith element of the jth frame has an index value a_{ij} and the corresponding RGB vector (R_{ij}, G_{ij}, and B_{ij}).

In order to embed secret data, a sender modifies a cover image $\mathbf{X} = \{x_1, x_2, \cdots, x_n\}$ to generate a stego image $\mathbf{Y} = \{y_1, y_2, \cdots, y_n\}$, in which n is the number of pixels in a cover. The generated additive distortion function $D(\mathbf{X}, \mathbf{Y})$ between $\mathbf{X}$ and $\mathbf{Y}$ is the sum of the embedding cost of each element, s.t.

$$D(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{n} \rho_i(x_i, y_i), \tag{1}$$

in which $\rho_i \in [0,\infty)$ is the embedding cost of changing x_i to y_i. According to the adjusted range I, the common embedding operations on x_i are divided into two types, i.e., the binary when $|I| = 2$ and the ternary when $|I| = 3$. For example, the embedding operations of ± 1 is the ternary embedding that is expressed by $I = \{-1, 0, +1\}$ and the stego $y_i = \{x_i - 1, x_i, x_i + 1\}$. We denote the embedding costs of three cases of $y_i = \{x_i - 1, x_i, x_i + 1\}$ as ρ^-_i, ρ_i, and ρ^+_i, respectively, in which $\rho_i = 0$ and $\rho^-_i, \rho^+_i \in (0,+\infty)$.

In [39], for a given embedding cost, the modification probability p_i can be obtained:

$$p_i^{(I)} = \frac{e^{-\lambda \rho_i^{(I)}}}{\sum_{I \in \{-1,0,+1\}} e^{-\lambda \rho_i^{(I)}}}, \tag{2}$$

where $\lambda \in [0,\infty)$ is obtained from the constraints of the modification probability and the m-bit secret data in (3).

$$H(p) = -\sum_{i=1}^{n} p_i \log p_i = m. \tag{3}$$

In [39], it has been theoretically proved that the minimum distortion is achievable when embedding m-bit secret data into n cover data.

$$D_{\min}(m, n, \rho) = \sum_{i=1}^{n} \rho_i p_i. \tag{4}$$

Derivation of the minimum distortion corresponding to a fixed embedding payload is the rate-distortion bound [39]. When embedding m-bit secret data, the sender should make the distortion between the cover image and the stego image as small as possible, which can be formulated as the following optimization problems:

$$\begin{aligned} &\min_{D_{\min}} D_{\min}(m, n, \rho), \\ &\text{subject to } H(p) = m. \end{aligned} \tag{5}$$

In the actual embedding process, we first define a suitable distortion function and then combine the coding methods, e.g., STC, to embed data and limit the distortion as far as possible.

For secret data $\mathbf{M} = (M_1, M_2, \cdots, M_m)^{\mathrm{T} \in \{0,1\}^m}$, we use STC to find the closest codewords to cover in coset C of $\mathbf{M}$ as stego.

$$\mathrm{Emb}(\mathbf{X}, \mathbf{M}) = \arg \min_{\mathbf{Y} \in C(\mathbf{M})} D(\mathbf{X}, \mathbf{Y}). \tag{6}$$

The definition of coset C is shown in (7).

$$C(\mathbf{M}) = \{\mathbf{Z} \in \{0, 1\}^n \mid \mathbf{HZ} = \mathbf{M}\}, \tag{7}$$

where $\mathbf{H} \in \{0, 1\}^{m \times n}$ is a parity-check matrix.

As the stego is found in the codeword $\mathbf{Z}$, the secret data can be extracted by multiplying the stego with $\mathbf{H}$, as shown in (8).

$$\mathbf{M} = \mathrm{Ext}(\mathbf{Y}) = \mathbf{HY}. \tag{8}$$

3. Proposed Method

In Figure 1, we show the framework of the proposed method for steganography in dynamic GIF images. First, we decompose dynamic GIF images into frames, each of which is a static GIF image. We design new distortion function for each frame. Based on the existing distortion functions, we obtain the initial embedding costs ${\rho^B}_{ij}$ for each frame. Meanwhile, according to the characteristics of the GIF image palette and the difference between the current frame and the adjacent frames in dynamic GIF image, we calculate adjustment factors ${\rho^C}_{ij}$ and ${\rho^V}_{ij}$ for initial distortion function to construct a new distortion function to get the improved embedding costs $\bar{\rho}_{ij}$.

If the payload contains m bits, we dynamically allocate the m bits to each frame. After identifying the distortion function and the payload for each frame, we obtain the stego GIF image by data embedding.

3.1. Distortion Function Initialization. We first design a steganography method for each frame of dynamic GIF images. By changing the index values in the index matrix, the secret data can be embedded into the cover with slight modification in the image content.

For a 256-color dynamic GIF image, each index value $a_{ij} \in [0, 255]$ corresponds to an RGB vector (R_{ij}, G_{ij}, and B_{ij}). We make an analogy between the RGB values of GIF images and the gray values of gray images. Thus, the algorithms of WOW, UNIWARD, HILL, etc. can be used to define the distortion function for each GIF frame.

For each frame, we extract the matrix of RGB values from the GIF image palette separately. We use the spatial distortion function, i.e. HILL and UNWARD as initial distortion function to calculate the embedding costs of each pixel in RGB channels, denote the embedding cost of the ith element in the jth frame in RGB channels as $\rho_{ij}(R)$, $\rho_{ij}(G)$, and $\rho_{ij}(B)$. Therefore, we obtain an initial distortion function

$$\rho_{ij}^B = \frac{\rho_{ij}(R) + \rho_{ij}(G) + \rho_{ij}(B)}{3}. \tag{9}$$

3.1.1. Palette Sort Algorithm. As shown in Figure 2, the GIF image palette is arranged out of order. Therefore, similar RGB vectors may be separated. For example, the difference between the RGB values corresponding to the index 2 and the index 100 is smaller than the difference between the RGB values corresponding to the index 2 and the index 3. To achieve better undetectability, the initial pixel value of the cover and the modified value of the stego should be as close as possible. Considering that Euclidean distance is the distance between two points in a multidimensional space, the RGB cube is a three-dimensional space. Hence, we rearrange the GIF image palette using the Euclidean distance as criterion.

We obtain the original palette $\boldsymbol{\Theta}$ and the original index matrix $\boldsymbol{\Omega}$ from the GIF image. The original palette contains 256 RGB vectors. We denote each element in the palette as $\theta_a = (R_a, G_a, B_a)$, in which a is index value, the range is 0 to 255.

As the example shown in Figure 3, we have $\theta_4 = (0.02, 0.03, 0.02)$. By rearranging the original palette, we can obtain a sorted palette $\bar{\Theta}$ and a new index matrix $\bar{\boldsymbol{\Omega}}$. We denote the new palette as $\overline{\theta}_{\bar{a}} = (R_{\bar{a}}, G_{\bar{a}}, B_{\bar{a}}), \bar{a} \in [0, 255]$. The steps of rearranging the palette are depicted as follows.

Step 1. We find l most frequently used RGB vectors from $\boldsymbol{\Omega}$ and denote their index value as $\boldsymbol{\Phi} = \{\phi 1, \cdots, \phi_l\}$. For each vector corresponding to the index in $\boldsymbol{\Phi}$, we find a vector from $\boldsymbol{\Theta}$ (*excluding the indexes in* $\boldsymbol{\Phi}$) that has the smallest distance. We denote the index values of these vectors as $\boldsymbol{\Phi}^{'} = \{\phi_1^{'}, \cdots, \phi_l^{'}\}$.

Step 2. We generate a random number a^{rand} from 0 to 255. Let θa^{rand} be the first row $\overline{\theta}_{\bar{a}}(\bar{a} = 0)$ in the new palette $\bar{\boldsymbol{\Theta}}$. Meanwhile, we find the positions in $\boldsymbol{\Omega}$ that have the index values equal to a^{rand}. On the same positions in $\bar{\boldsymbol{\Omega}}$, we set the

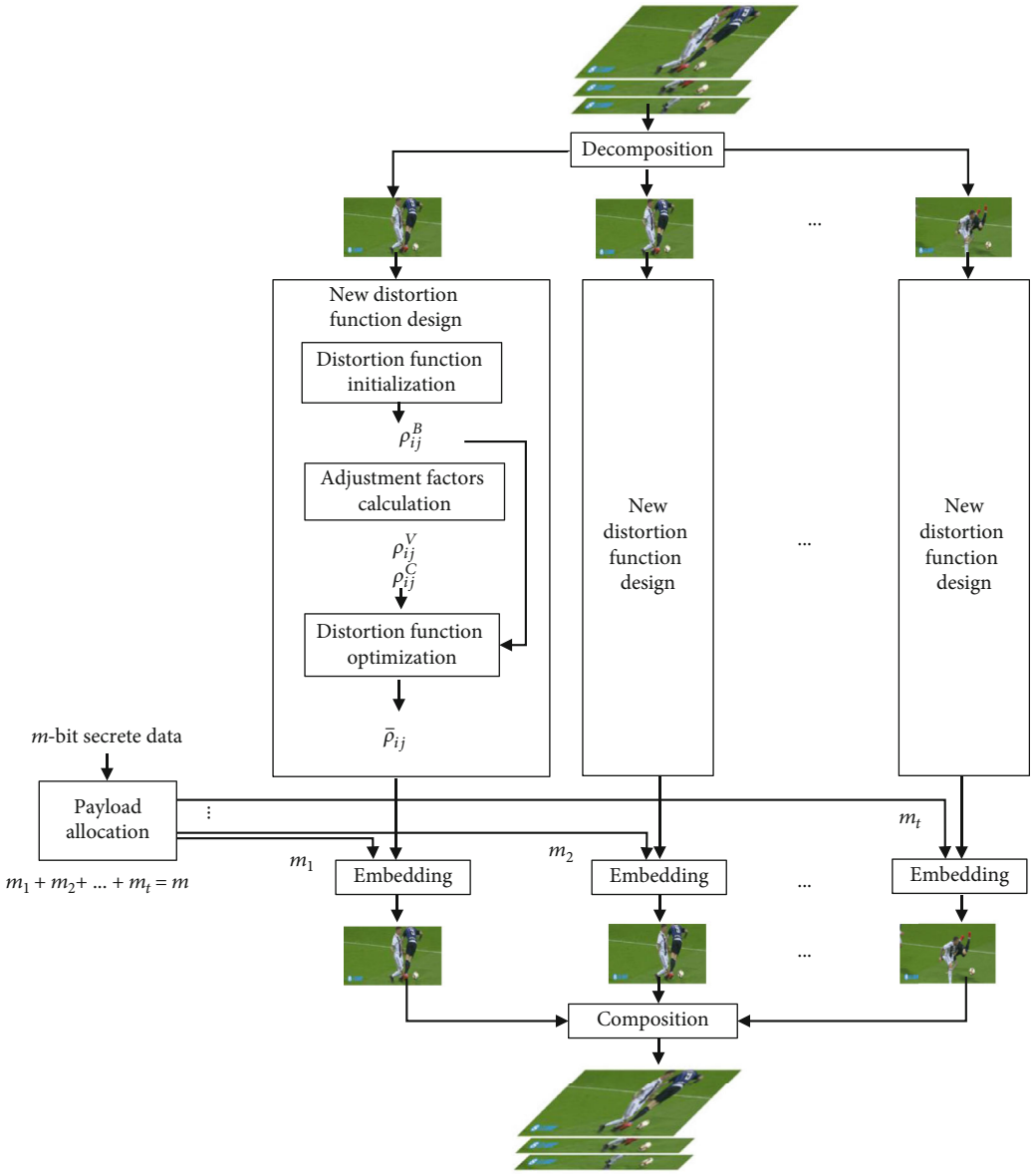

Figure 1: The framework of dynamic GIF image steganography.

Index value	Actual color value		
	R	G	B
0	0	0	0
1	0.9765	0.9621	0.9822
2	0.0196	0.0211	0.0185
3	0.3254	0.3341	0.3550
...	...	...	...
100	0.0157	0.0148	0.0157
...	...	...	...
255	1	1	1

Index matrix: 0 ... 3 / ⋮ ⋱ ⋮ / 87 ... 255

Figure 2: Demonstration of the palette format.

values as $\bar{a}$. As the example in Figure 3, since $a^{\text{rand}} = 4$, we assign θ_4 as $\overline{\theta}_0$. We find the positions in $\boldsymbol{\Omega}$ that have the index values 4 and set the values on these positions in $\bar{\boldsymbol{\Omega}}$, as 0.

Step 3. From the palette $\boldsymbol{\Theta}$, we find a vector θ_{a^*} that has the smallest Euclidean distance to $\overline{\theta}_{\bar{a}}$. The vector θ_{a^*} must have not been used in previous steps. We set this vector as $\overline{\theta}_{\bar{a}+1}$ in the new palette $\bar{\boldsymbol{\Theta}}$. We further find the positions in $\boldsymbol{\Omega}$ that have the index values equal to $a*$. On the same positions in $\bar{\boldsymbol{\Omega}}$, we set the values as $\bar{a}+1$. As the example in Figure 3, since θ_4 has the smallest Euclidean distance with θ_{56}, we assign θ_{56} as $\overline{\theta}_1$. Meanwhile, we find the positions in $\boldsymbol{\Omega}$ that have the index values 56 and set the values on these positions in $\bar{\boldsymbol{\Omega}}$ as 1.

Step 4. If the index value $a*$ of θ_{a^*} belongs to the set $\boldsymbol{\Phi}$ or $\boldsymbol{\Phi}'$, say $\theta_{\phi'_l}$ or θ_{ϕ_l}, we must use θ_{ϕ_l} or $\theta_{\phi'_l}$ to fill the $(\bar{a}+2)$-th vector in $\bar{\boldsymbol{\Theta}}$. Otherwise, we use Step 3 to fill the $(\bar{a}+2)$-th vector in $\bar{\boldsymbol{\Theta}}$. As shown in Figure 3, as $a* = 56$ and $56 = \phi'_1 \in \boldsymbol{\Phi}'$, we

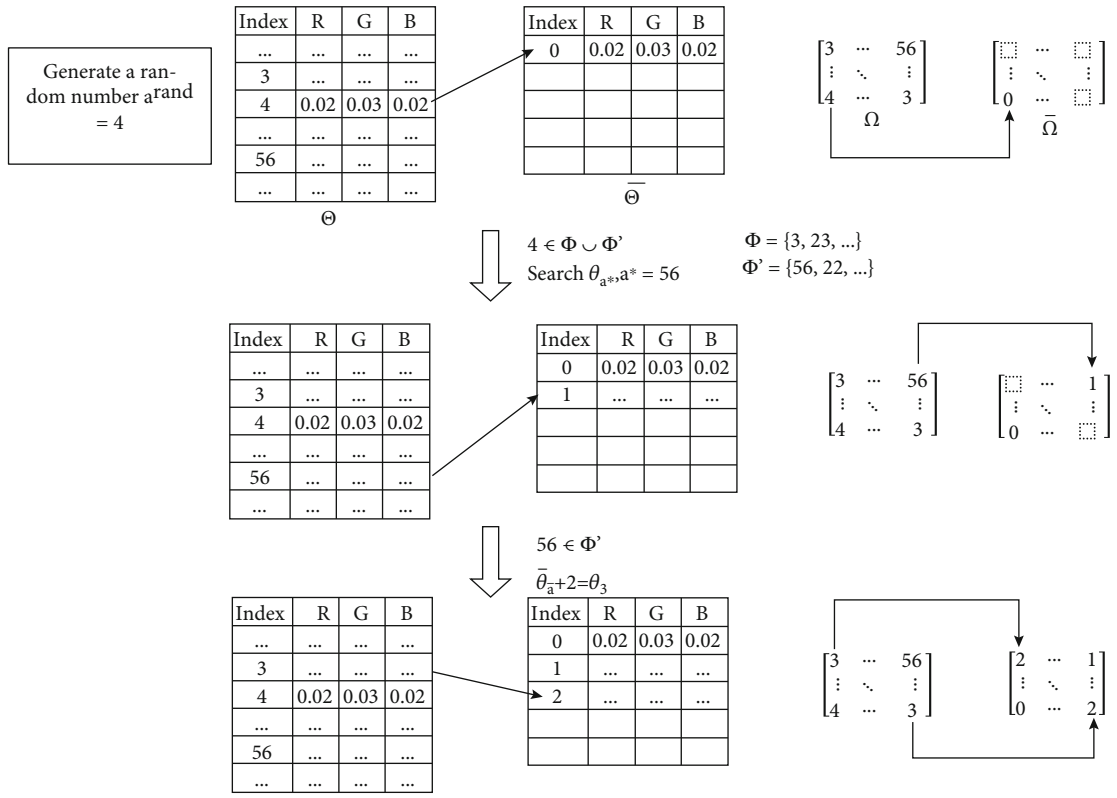

Figure 3: The process of rearranging palette.

use θ_3 ($\phi_1 = 3$) to fill the $(\bar{a}+2)$-th vector in $\bar{\mathbf{\Theta}}$. Meanwhile, we set the values in corresponding positions in $\bar{\mathbf{\Omega}}$ as 3.

3.2. Distortion Function Optimization Based on Magnitude. In dynamic GIF image steganography, we embed data by modifying the index values. It is different from the steganography method for spatial images that embeds data by modifying pixel values. However, we can use the STC framework by optimizing the distortion functions defined for spatial images. Before using the ±1 embedding algorithm, we initialize the distortion function by state-of-the-art designs, e.g., WOW, UNIWARD, or HILL.

For WOW, the distortion algorithm calculates a weighted difference *embedding suitability* to get the embedding costs of this pixel, using the difference between the residual $\mathbf{R}$ of the cover image and the residual $\mathbf{R}_i$ that only the ith pixel is modified, as shown in (10),

$$\xi_{ij}^{(k)} = \left|\mathbf{R}^{(k)}\right| * \left|\mathbf{R}^{(k)} - \mathbf{R}_i^{(k)}\right|^{\curvearrowleft}, \tag{10}$$

in which "$*$" is the convolution operator, "$\curvearrowleft$" is an operator of rotating 180 degrees, and $\xi^{(k)}{}_i$ is the embedding suitability, $|\cdot|$ is the absolute operator. Besides, $|\mathbf{R}^{(k)} - \mathbf{R}^{(k)}{}_i|$ is calculated by

$$\left|\mathbf{R}^{(k)} - \mathbf{R}_i^{(k)}\right| = \left|\mathbf{K}^{(k)} * \mathbf{X} - \mathbf{K}^{(k)} * \mathbf{X}_i\right| = \left|\mathbf{K}^{(k)} * \mathbf{X}_i'\right|, \tag{11}$$

where $\mathbf{K}$ is a filter, $\mathbf{K}^{(k)}$ the kth filter, and $\mathbf{X}_i$ is the matrix that only the ith pixel is modified, $\mathbf{X}^{'}{}_i$ is the matrix with the same size as the cover image $\mathbf{X}$. When the modified amplitude is 1, the ith element of $\mathbf{X}^{'}{}_i$ is 1, while the others are all 0. The value of the weight difference of WOW depends on the modified amplitude and the filter. Since the calculation of the weight difference uses an absolute value, changing the pixel value in the forward direction (the embedding operation of +1) and changing the pixel value in reverse (the embedding operation of -1) have no distinction on the result.

However, when embedding data in GIF, we must modify the index values. The modification would be result in the change of the RGB values. More importantly, the modifications of +1 and -1 have different impacts on the RGB values. As shown in Figure 4, the index value of x_5 is 5 and the corresponding R value is 0.42. When adding 1 to x_5, the index value changes to 6, the corresponding R value changes to 0.26, and the modification amplitude is 0.16. When subtracting 1 from x_5, the index value changes to 4, the corresponding R value changes to 0.41, and the modification amplitude is 0.01. For x_6 with index value equal to 6, the corresponding R value is 0.26 and the modification amplitude is 0.16 when subtracting 1 from x_6. Different operations on different index values would have different modification amplitudes on RGB values. Therefore, when hiding data into GIF, we should set different embedding costs for different operations on different index values.

According to this point, we calculate the modification amplitudes on RGB values when adjusting ±1 on index values and optimize the initial distortion function according to the modified amplitude. The pixels with less impact are

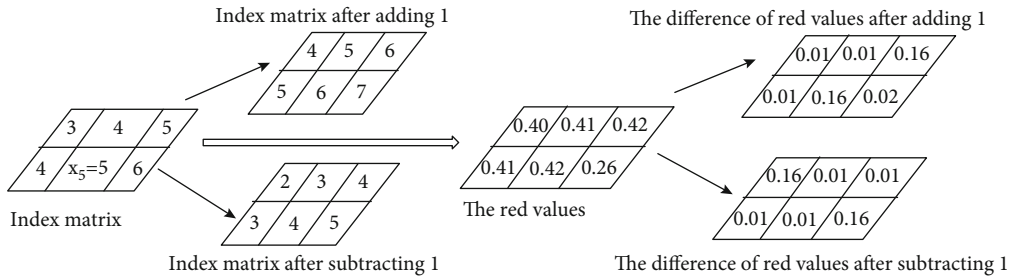

Figure 4: An example for different modified amplitude.

assigned with small embedding costs, and the pixels with more impacts are assigned with large embedding costs.

Accordingly, we propose to include an adjustment parameter δ called amplitude weight to manipulate the optimized proportion of the modification amplitude in the new distortion function. The setting of δ will be discussed in Section 4.1. Since the initial distortion functions are define for RGB channels, we provide the adjustment factors for three channels, respectively. In (12), ${\rho^{C+}}_{ij}(R)$, ${\rho^{C+}}_{ij}(G)$, and ${\rho^{C+}}_{ij}(B)$ are the optimization factors for the +1 embedding operations of the ith pixel in the jth frame in three color channels and ${\rho^{C-}}_{ij}(R)$, ${\rho^{C-}}_{ij}(G)$, and ${\rho^{C-}}_{ij}(B)$ are the optimization factors for the -1 embedding operations of ith pixel in the jth frame in three color channels. δ is the adjustment parameter called amplitude weight. ${r^+}_{ij}$, ${g^+}_{ij}$, and ${b^+}_{ij}$ are the absolute difference of the RGB values of a_{ij} and $(a_{ij}+1)$ and ${r^-}_{ij}$, ${g^-}_{ij}$, and ${b^-}_{ij}$ are the absolute difference of the RGB values of a_{ij} and $(a_{ij}-1)$.

$$\begin{cases} \rho_{ij}^{C+}(R) = \delta \times {r^+}_{ij} + \dfrac{255-\delta}{255}, \\ \rho_{ij}^{C-}(R) = \delta \times {r^-}_{ij} + \dfrac{255-\delta}{255}, \\ \rho_{ij}^{C+}(G) = \delta \times {g^+}_{ij} + \dfrac{255-\delta}{255}, \\ \rho_{ij}^{C-}(G) = \delta \times {g^-}_{ij} + \dfrac{255-\delta}{255}, \\ \rho_{ij}^{C+}(B) = \delta \times {b^+}_{ij} + \dfrac{255-\delta}{255}, \\ \rho_{ij}^{C-}(B) = \delta \times {b^-}_{ij} + \dfrac{255-\delta}{255}. \end{cases} \tag{12}$$

Combining three optimization factors with the initial distortion functions, we adjust the distortion function as (13).

$$\begin{aligned} \rho_{ij}^{T+} &= \frac{\rho_{ij}^{C+}(R)\rho_{ij}(R) + \rho_{ij}^{C+}(R)\rho_{ij}(G) + \rho_{ij}^{C+}(R)\rho_{ij}(B)}{3}, \\ \rho_{ij}^{T-} &= \frac{\rho_{ij}^{C-}(R)\rho_{ij}(R) + \rho_{ij}^{C-}(R)\rho_{ij}(G) + \rho_{ij}^{C-}(R)\rho_{ij}(B)}{3}. \end{aligned} \tag{13}$$

3.3. Distortion Function Improvement Using the Correlation of Interframes. In spatial image steganography, the texture and edge regions are more appropriate for data hiding than the other regions, as it is more secure against the steganalysis. Most STC embedding methods assign these regions with low embedding costs. This principle can also be used for steganography in dynamic GIF images. As there are many frames in a dynamic GIF image, we also use the changes in interframes. When displaying animation effects, most of the successive frames have strong correlations, which can be found in Figure 5.

We propose to integrate the correlations of interframes into distortion function optimization. The pixels that have larger changes in interframes would be assigned with smaller embedding costs. First, we calculate the motion differences of adjacent frames of each pixel. Then, we select the appropriate pixels according to these motion differences. After that, we set adjustment parameters to adjust the embedding costs of these pixels to achieve the adjustment factor ${\rho^V}_{ij}$.

For each pixel in dynamic GIF image **A**, we calculate the difference between each frame and the next frame by (14).

$$\Delta d_{ij} = \sqrt{\left(\Delta R_{ij}\right)^2 + \left(\Delta G_{ij}\right)^2 + \left(\Delta B_{ij}\right)^2}, \tag{14}$$

where Δd_{ij} is the difference between the ith pixel in the jth frame and the ith pixel in the $(j+1)$th frame. ΔR_{ij} is the absolute difference of R_{ij} and $R_{i(j+1)}$, ΔG_{ij} is the absolute difference of G_{ij} and $G_{i(j+1)}$, ΔB_{ij} is the absolute difference of B_{ij} and $B_{i(j+1)}$, $i \in [1, n]$, $j \in [1, t]$. n is the number of pixels in a frame and t is the number of frames in dynamic GIF image.

As the adjacent pixels are related, we use the average value of the differences of the pixels in a 3×3 block as the difference of the pixel in the center of the block. In order to better evaluate the degree of change in inter frames, we average the difference between the present and the previous frames and the difference between the present and the next frames. We use the average as the motion difference of adjacent frames, which is depicted in (15),

$$d_{ij} = \begin{cases} \Delta d_{ij}, & j = 1, \\ \dfrac{\Delta d_{i(j-1)} + \Delta d_{ij}}{2}, & \text{otherwise}, \\ \Delta d_{ij}, & j = t, \end{cases} \tag{15}$$

Figure 5: Three consecutive frames in the fifth dynamic GIF image in the dataset.

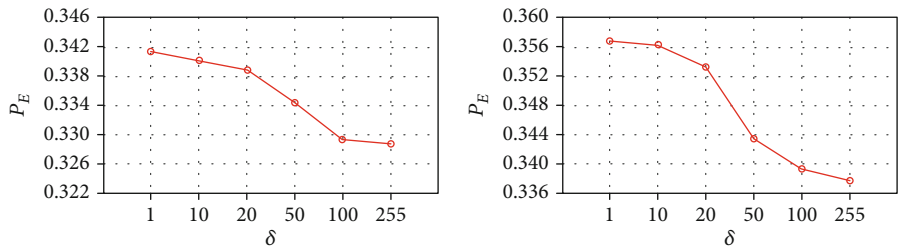

Figure 6: Testing error with respect to δ. (a) Gp-HILL in SPORTBase. (b) Gp-UNIWARD in SPORTBase.

in which d_{ij} is the motion difference of the ith pixel in the jth frame, and Δd_{ij} is the difference between the ith pixel in the j th frame and the ith pixel in the $(j+1)$th frame.

After calculating the interchanges of each pixel of the dynamic GIF image, we sort all pixels according to the motion difference and select the pixels with the top a% of large changes to decrease their embedding costs. Considering that changes exceeding 38 dB are imperceptible to the human eye, we combine the image content to select the pixels with changes below 38 dB to calculate the selection range adaptively.

$$\alpha = \beta(d_M - d_T), \tag{16}$$

where is d_M the mean of the motion differences of the pixels with changes below 38 dB, d_T the motion differences of the pixels with changes equal 38 dB, and the motion difference parameter β is empirically set as 0.2. We select the pixels with the top α% by (17)

$$v_{ij} = \begin{cases} d_{ij}, & \text{if the } i\text{th pixel in the } j\text{th frame} \in \alpha\%, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

Subsequently, we define the adjustment factor in (18) according to the correlation of interframes.

$$\rho_{ij}^{V} = e^{-v_{ij}}. \tag{18}$$

Finally, we obtain the final distortion function

$$\begin{aligned} \bar{\rho}^{+}{}_{ij} &= \rho_{ij}^{V}\rho_{ij}^{C+}, \\ \bar{\rho}^{-}{}_{ij} &= \rho_{ij}^{V}\rho_{ij}^{C-}. \end{aligned} \tag{19}$$

3.4. Payload Allocation. In dynamic GIF image steganography, we calculate the embedding costs for each frame. There are more pixels with small embedding cost in some frames. Therefore, these frames can accommodate more secret data. In order to achieve a better security with the condition of a constant total payload, we assign each frame with different payload according to the characteristics of each frame itself.

According to the calculation of the initial embedding costs and optimization factors, we get the optimized embedding costs $\bar{\rho}_{ij}$ for each pixel in dynamic GIF image. During data hiding, the whole distortion should satisfy the rate distortion bound in (20),

$$\begin{aligned} &\min_{D_{\min}} D_{\min}(m, n*t, \bar{\rho}) = \sum_{i=1}^{n}\sum_{j=1}^{t}\bar{\rho}_{ij}\bar{p}_{ij}, \\ &\text{subject to } \sum_{i=1}^{n}\sum_{j=1}^{t}H(\bar{p}) = m, \end{aligned} \tag{20}$$

where n is the number of pixels in a frame and t is the number of frames in dynamic GIF image. $\bar{\rho}_{ij}$ is the embedding cost of the ith pixel in the jth frame. $\bar{p}_{ij}$ is the modification probability of the ith pixel in the jth frame. $H(\bar{p})$ is a entropy function to calculate the payload based on the modification probability, which is depicted in (3). The modification probability $\bar{p}_{ij}$ can be obtained according to the embedding cost $\bar{\rho}_{ij}$ and a parameter λ. The parameter λ is obtained from the constraints of the modification probability and the payload of secret data in (20).

After obtaining the distortion function, we input the embedding costs of all frames and the payload of secret data into the constraints in (20) to calculate the parameter λ.

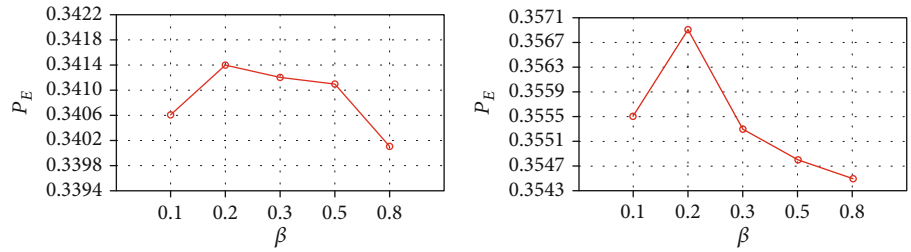

Figure 7: Testing error with respect to β. (a) Gp-HILL in SPORTBase. (b) Gp-UNIWARD in SPORTBase.

Figure 8: Comparison of stego images. (a) The original frame. (b) Stego generated by initial distortion function. (c) The stego by Shi et al. [37]. (d) The stego by us.

Table 1: Comparison of the PSNR results between the basic method using HILL, Shi's method using HILL, and our GP-HILL in SPORTBase.

Methods	Payload (bpp)				
	0.05	0.1	0.15	0.2	0.25
Basic (HILL)	40.51	37.00	34.90	33.37	32.18
Shi et al.'s (HILL)	31.83	30.79	29.70	28.60	27.59
Gp-HILL	45.71	42.13	40.01	38.46	37.27

Accordingly, we obtain the modification probability of each frame based on the embedding costs and the parameter λ. Subsequently, embedding payload of each frame can be calculated by (21)

$$m_j = \sum_{i=1}^{n} H\left(\bar{p}_{ij}\right), j \in [1, t]. \quad (21)$$

4. Experimental Results

To verify the proposed method, we have conducted many experiments. Two dynamic GIF image datasets are constructed, i.e., the SPORTBase and the HAPPYBase. Each database contains 500 dynamic GIF images, and each image has 20 frames, i.e., 10,000 static GIF images in each dataset. They are all color dynamic images with 8-bit index values. The SPORTBase are football games downloaded from https://www.zhibo8.cc/. The HAPPYBase are emoticons from https://www.soogif.com/, https://www.sina.com.cn/ and other websites. Both are available in https://github.com/jzlin1997/GIF-Image-Steganography.

We use HILL, UNIWARD, and WOW as the initial distortion function in the proposed method. After optimizing the initial distortion function and allocating the embedding payloads, we generate a new steganography method for dynamic GIF images. We name the proposed steganography based on improved HILL, UNIWARD and WOW as Gp-HILL, Gp-UNIWARD and Gp-WOW, respectively.

We use the steganalysis tools of the state-of-the-art SPAM feature set [19] and SRM feature set [18] with the ensemble classifiers [22]. We obtain the RGB values of stego GIF images to calculate the gray values and extract SPAM features from the gray values. Half of the cover and stego feature sets are used for training, and the rest are for testing. We use the minimal total error P_E with equal priors achieved on

Table 2: Comparison of the steganalysis results in SPORTBase.

Feature	Methods	Payload (bpp) 0.05	0.1	0.15	0.2	0.25
SPAM	Basic (HILL)	0.4360	0.3664	0.2949	0.2273	0.1672
	Shi et al.'s (HILL)	0.3674	0.2810	0.2039	0.1418	0.1034
	Ours (HILL)	0.4516	0.3963	0.3414	0.2875	0.2367
	Basic (UNIWARD)	0.4455	0.3829	0.3199	0.2543	0.1932
	Shi et al.'s (UNIWARD)	0.4013	0.3161	0.2429	0.1766	0.1281
	Ours (UNIWARD)	0.4563	0.4064	0.3569	0.3019	0.2489
	Basic (WOW)	0.4362	0.3633	0.2934	0.2280	0.1718
	Shi et al.'s (WOW)	0.4043	0.3249	0.2507	0.1795	0.1292
	Ours (WOW)	0.4495	0.3910	0.3341	0.2803	0.2267
SRM	Basic (HILL)	0.1723	0.0437	0.0107	0.0039	0.0018
	Shi et al.'s (HILL)	0.0961	0.0440	0.0318	0.0287	0.0272
	Ours (HILL)	0.2356	0.0982	0.0395	0.0165	0.0077
	Basic (UNIWARD)	0.1950	0.0498	0.0125	0.0036	0.0017
	Shi et al.'s (UNIWARD)	0.0922	0.0468	0.0311	0.0265	0.0238
	Ours (UNIWARD)	0.2771	0.1123	0.0433	0.0175	0.0071
	Basic (WOW)	0.1804	0.0441	0.0105	0.0036	0.0016
	Shi et al.'s (WOW)	0.0944	0.0465	0.0316	0.0260	0.0242
	Ours (WOW)	0.2616	0.1080	0.0410	0.0180	0.0085

the testing sets as the criterion to evaluate the performances of the feature sets. In (22), P_{FA} is the false alarm rate and P_{MD} is the missed detection rate. The average of P_E by 10 random tests is used to evaluate the performance.

$$P_E = \min_{P_{FA}} \left(\frac{P_{FA} + P_{MD}}{2} \right). \tag{22}$$

In Section 4.1, we discuss the adjustment parameter, i.e., the amplitude weight. Based on appropriate amplitude weight, we find a suitable value of the motion difference parameter. Subsequently, we compare our proposed method with the basic method using initial distortion functions and Shi et al. [37] in Section 4.2 and Section 4.3. We give a subjective evaluation of the stego image in Section 4.2 and a specific analysis of undetectability in Section 4.3.

4.1. Parameter Determination. In Section 3.2, we have analyzed that the modification amplitude is an important factor for distortion function. Considering that the modification amplitudes on pixel value of different operations are different, we not only embed the data in texture regions but also embed the data in the pixels with small modification amplitude. We define an adjustment parameter δ called amplitude weight to adjust the optimized proportion of the modification amplitude in the new distortion function. If δ is too large, the embedding costs of the pixels with small modification but in smooth region would become much smaller. It will ignore the effects of embedding in texture regions.

We use Gp-HILL and Gp-UNIWARD to embed 0.15 bpp into dynamic GIF images in SPORTBase with different values of δ. The SPAM errors are shown in Figures 6(a) and 6(b). The results indicate that the largest testing errors can be achieved when $\delta = 1$. Therefore, we set $\delta = 1$.

Based on the identified amplitude weight δ, we look for the optimal parameter β of motion difference in the SPORTBase. We use Gp-HILL and Gp-UNIWARD to embed 0.15 bpp into GIF images and vary the values of β, respectively. The results shown in Figure 7 indicate that the largest testing errors can be achieved when $\beta = 0.2$.

4.2. Image Quality. In Figure 8, we compare quality of the stego frames by the proposed method and the other methods. We use HILL as the initial distortion function. The stegos are generated by the initial function, the function in [37] and ours. The embedding payload is 0.2 bpp. Figure 8(a) shows the original frame. Figures 8(b)–8(d) shows the stegos generated by different distortion functions. Traditional distortion functions used for GIF embedding only consider the embedding in texture regions, which always ignore the differences of modification amplitude. Therefore, there are some drawbacks like the black points appearing in the blue coat in Figure 8(b). However, the distortion function in [37] focuses on embedding secret data into the pixels with a smaller modification amplitude, a large piece of white in the blue coat in Figure 8(c). The results show that the proposed method achieves a better quality.

Meanwhile, based on the HILL, we calculate the mean PSNR (peak-signal-to-noise ratio) of the stego generated by initial distortion function, the stego by Shi et al. and our Gp-HILL in Table 1. As shown in Table 1, all the PSNR of the stego generated by Gp-HILL has been improved. For example, the PSNR of the stego generated by Gp-HILL is 40.01 when the embedding payload is 0.15 bpp. Compared with the PSNR of the stego generated

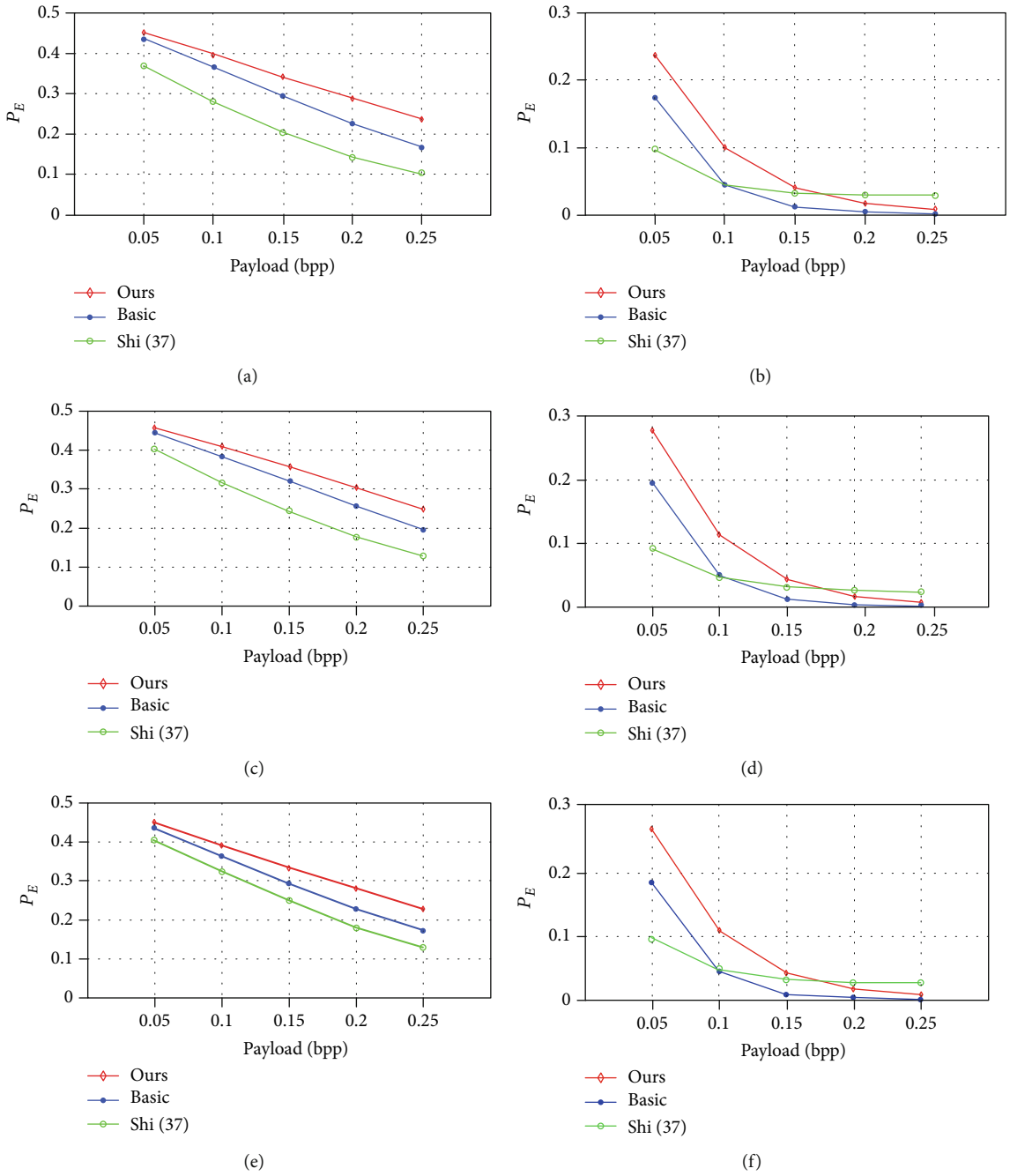

Figure 9: Comparison of the steganalysis results in SPORTBase. (a) Gp-HILL with SPAM. (b) Gp-UNIWARD with SRM. (c) Gp-UNIWARD with SPAM. (d) Gp-UNIWARD with SRM. (e) Gp-WOW with SPAM. (f) Gp-WOW with SRM.

by the basic method using HILL, our method can improve 5.11. Moreover, from 0.05 bpp to 0.2 bpp, the PSNR of the stego generated by Gp-HILL has reached an undetectable level by the human eye.

4.3. Security against Steganalysis. We have also conducted many experiments to verify the capabilities of countering steganalysis. The proposed Gp-HILL, Gp-UNIWARD, and Gp-WOW are used to hide secret data into the dynamic GIF images in SPORTBase and HAPPYBase. Five different payloads from 0.05 bpp to 0.25 bpp are used.

We use SPAM and SRM to extract features from covers and stegos to evaluate the security of the proposed method. The proposed method is compared with the basic method, Shi et al.'s method in [37]. The basic method and Shi et al.'s method use HILL, UNIWARD, and WOW as initial distortion function.

In Table 2 and Figures 9(a)–9(f), we show the steganalysis results of data hiding in SPORTBase. Table 2 indicated that the proposed method outperforms the basic methods and the Shi et al.'s method using three distortion functions with SPAM feature. For example, the testing error of Gp-

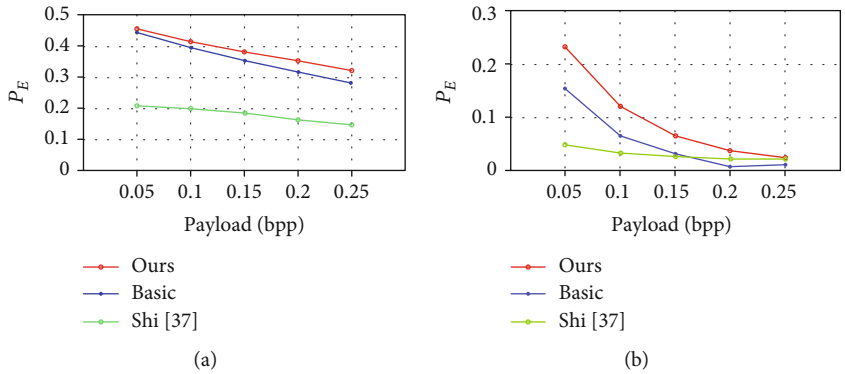

Figure 10: Comparison of the steganalysis results in HAPPYBase. (a) Gp-HILL with SPAM. (b) Gp-UNIWARD with SRM.

Table 3: Comparison of the steganalysis results in HAPPYBase.

Feature	Methods	Payload (bpp) 0.05	0.1	0.15	0.2	0.25
SPAM	Basic (HILL)	0.4473	0.3956	0.3545	0.3180	0.2829
	Shi et al.'s (HILL)	0.2080	0.2007	0.1862	0.1654	0.1488
	Ours (HILL)	0.4547	0.4160	0.3835	0.3538	0.3238
SRM	Basic (HILL)	0.1544	0.0662	0.0320	0.0184	0.0119
	Shi et al.'s (HILL)	0.0490	0.0333	0.0260	0.0225	0.0213
	Ours (HILL)	0.2326	0.1212	0.0659	0.0380	0.0246

HILL with SPAM is 0.3414 when the embedding payload is 0.25 bpp. It can improve 0.0465 comparing the testing error of the basic method with SPAM using HILL. With the increase of embedding payload, the improvement of testing error also increases. Further, Table 2 shows all the testing errors obtained by the proposed method with SRM are higher than the basic method from 0.05 bpp to 0.5 bpp and are higher than the Shi et al.'s method from 0.05 bpp to 0.15 bpp. When payload becomes larger, embedding in texture area has lost its effect, such as the testing error of Gp-HILL with SRM is 0.0018 when the embedding payload is 0.25 bpp, therefore, the testing errors obtained by the proposed method with SRM are lower than the Shi et al.'s method. But the image quality of Shi et al.'s method is worse. It is shown the proposed steganography method could obtain improvements in the performance of steganalysis.

To further evaluate the performance of the proposed method, we do experiments using HAPPYBase. Figures 10(a) and 10(b) and Table 3 shows the comparisons of testing errors between Gp-HILL, the basic method, and Shi et al.'s method using HILL in HAPPYBase with SPAM and SRM. As shown in Table 3, the testing errors of the proposed method are higher than the basic method and the Shi et al.'s method in all situation in HAPPYBase, which has richer texture than SPORTBase. The results indicate the proposed method for dynamic GIF images steganography could achieve better performance on resisting the modern steganalysis and verify that the parameters are also appropriate for other databases.

5. Conclusions

The paper presents a new steganography method for the dynamic GIF images based on STC framework. We rearrange the palette and calculate the modification amplitudes on pixel values according to the mapping relationships between index values and RGB values. According to the modification amplitudes on pixel values, we calculate the adjustment factors for each color channel. Considering the strong correlation of interframes, we use the motion difference of adjacent frames as an adjustment factor to adjust embedding costs. Combining adjustment factors, we design a new method of distortion function specification. Besides, with the embedding probabilities on different frames, we also assign different payloads for each frame to achieve higher security. The experimental results show that the proposed method has a better performance than previous works.

Data Availability

The link of database and experiment results used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant 61572308, U1736213, and U1636206) and Shanghai Excellent Academic Leader Plan (16XD1401200).

References

[1] Z. Wang and X. Zhang, "Secure cover selection for steganography," *IEEE Access*, vol. 7, pp. 57857–57867, 2019.

[2] Z. Wang, Z. Qian, X. Zhang, M. Yang, and D. Ye, "On improving distortion functions for JPEG steganography," *IEEE Access*, vol. 6, pp. 74917–74930, 2018.

[3] Z. Wang, X. Zhang, and Z. Yin, "Joint cover-selection and payload allocation by steganographic distortion optimization," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1530–1534, 2018.

[4] Z. Zhou, Y. Mu, and Q. M. J. Wu, "Coverless image steganography using partial-duplicate image retrieval," *Soft Computing*, vol. 23, no. 13, pp. 4927–4938, 2019.

[5] F. A. P. Petitcolas, R. J. Anderson, and M. G. KUHN, "Information hiding a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[6] A. Westfeld, "F5A steganographic algorithm," in *Proceeding of 4th International Workshop on Information Hiding*, pp. 289–302, Pittsburgh, PA, USA, April 2001.

[7] V. Sedighi, J. Fridrich, and R. Cogranne, "Content-adaptive pentary steganography using the multivariate generalized Gaussian cover model," in *Proceeding of International Society for Optics and Photonics*, pp. 9409–94090H, San Francisco, CA, USA, March 2015.

[8] T. Filler and J. Fridrich, "Design of adaptive steganographic schemes for digital images," in *Proceeding of SPIE, Electronic Imaging, Media Watermarking, Security and Forensics of Multimedia XIII*, San Francisco, CA, USA, January 2011.

[9] Z. Wang, J. Lv, Q. Wei, and X. Zhang, "Distortion function for spatial image steganography based on the polarity of embedding change," in *Proceeding of the 15th International Workshop on Digital forensics and Watermarking (IWDW2016)*, pp. 487–493, Beijing, China, September 2016.

[10] W. Zhou, W. Zhang, and N. Yu, "A new rule for cost reassignment in adaptive steganography," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2654–2667, 2017.

[11] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[12] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in *Proceeding of SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX*, pp. 2-3, San Jose, CA, USA, Febuary 2007.

[13] T. Pevny, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proceeding of 12th International Workshop on Information Hiding*, pp. 161–177, Heidelberg, Calgary, AB, Canada, June 2010.

[14] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proceeding of IEEE Workshop on Information Forensic and Security*, pp. 234–239, Heidelberg, Germany, December 2012.

[15] V. Holub and J. Fridrich, "Digital image steganography using universal distortion," in *Proceedings of the first ACM workshop on Information hiding and multimedia security - IH&MMSec '13*, pp. 59–68, New York, NY, USA, June 2013.

[16] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 4206–4210, Paris, France, October 2014.

[17] L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient JPEG steganography," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 814–825, 2014.

[18] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[19] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[20] V. Holub and J. Fridrich, "Low complexity features for JPEG steganalysis using undecimated DCT," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, 2015.

[21] X. F. Song, F. L. Liu, C. F. Yang, X. Y. Luo, and Y. Zhang, "Steganalysis of adaptive JPEG steganography using 2D Gabor filters," in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security - IH&MMSec '15*, pp. 15–23, New York, NY, USA, June 2015.

[22] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[23] M. Kwan, *The gifshuffle home page[EB/OL]*, 2010, http://www.darkside.com.au/gifshuffle/.

[24] J. Fridrich and R. Du, "Secure steganographic methods for palette images," in *Proceeding of 3rd International Workshop Information Hiding*, pp. 47–60, Dresden, Germany, 1999.

[25] X. Zhang, S. Wang, and Z. Zhou, "Multibit assignment steganography in palette images," *IEEE Signal Processing Letters*, vol. 15, pp. 553–556, 2008.

[26] J. Fridrich, *A new steganographic method for palette-based images*, PICS, 1999.

[27] Z. R. Zhou, Z. C. Ji, Y. Z. Wang, and J. J. Lin, "A new algorithm of steganography based on palette image," in *2006 1ST IEEE Conference on Industrial Electronics and Applications*, pp. 1–4, Singapore, Singapore, May 2006.

[28] S. G. K. D. N. Samaratunge, "New steganography technique for palette based Iimages," in *2007 International Conference on Industrial and Information Systems*, pp. 335–339, Penadeniya, Sri Lanka, August 2007.

[29] S. M. Kim, Z. Q. Cheng, and K. Y. Yoo, "A new steganography scheme based on an index-color image," in *2009 Sixth International Conference on Information Technology: New Generations*, pp. 376–381, Las Vegas, Nevada, April 2009.

[30] I. T. Fathurohman, T. W. Purboyo, and R. A. Nugrahaeni, "Comparative analysis of steganography using LSB and adaptive method on GIF image," *International Journal of Applied Engineering Research*, vol. 12, no. 21, pp. 10999–11006, 2017.

[31] R. Munir, "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*, pp. 80–84, Bandung, Indonesia, October 2015.

[32] R. Munir, "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*, pp. 58–62, Balikpapan, Indonesia, October 2016.

[33] M. T. Juzar and R. Munir, "Message hiding in animated GIF using multibit assignment method," in *2016 International Symposium on Electronics and Smart Devices (ISESD)*, pp. 225–229, Bandung, Indonsia, November 2016.

[34] R. Munir, "Visual cryptography of animated GIF image based on XOR operation," in *2017 International Conference on Advanced Computing and Applications (ACOMP)*, pp. 117–121, Ho Chi Minh City, Vietnam, December 2017.

[35] R. K. Basak, K. Dasgupta, and P. Dutta, "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)*, pp. 248–252, Kolkata, India, November 2018.

[36] I. M. Saleh and H. H. Merzah, "Efficient data hiding system using LZW cryptography and gif image steganography," *International Journal of Technical Research and Applications*, vol. 3, no. 2, pp. 28–32, 2015.

[37] L. Shi, Z. Wang, Z. Qian, N. Huang, P. Puteaux, and X. Zhang, "Distortion function for emoji image steganography," *Computers, Materials & Continua*, vol. 59, no. 3, pp. 943–953, 2019.

[38] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 685–696, 2020.

[39] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 705–720, 2010.