Computer Modeling in Engineering & Sciences
DOI:10.32604/cmes.2020.010636
Article

Multiple Images Steganography of JPEG Images Based on Optimal Payload Distribution

Yang Pei[1,2], Xiangyang Luo[1,2,*], Yi Zhang[2] and Liyan Zhu[2]

[1]Zhong Yuan Network Security Research Institute, Zhengzhou University, Zhengzhou, 450000, China
[2]State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, 450000, China
[*]Corresponding Author: Xiangyang Luo. Email: luoxy_ieu@sina.com
Received: 15 March 2020; Accepted: 04 June 2020

Abstract: Multiple images steganography refers to hiding secret messages in multiple natural images to minimize the leakage of secret messages during transmission. Currently, the main multiple images steganography algorithms mainly distribute the payloads as sparsely as possible in multiple cover images to improve the detection error rate of stego images. In order to enable the payloads to be accurately and efficiently distributed in each cover image, this paper proposes a multiple images steganography for JPEG images based on optimal payload redistribution. Firstly, the algorithm uses the principle of dynamic programming to redistribute the payloads of the cover images to reduce the time required in the process of payloads distribution. Then, by reducing the difference between the features of the cover images and the stego images to increase the detection error rate of the stego images. Secondly, this paper uses a data decomposition mechanism based on Vandermonde matrix. Even if part of the data is lost during the transmission of the secret messages, as long as the data loss rate is less than the data redundancy rate, the original secret messages can be recovered. Experimental results show that the method proposed in this paper improves the efficiency of payloads distribution compared with existing multiple images steganography. At the same time, the algorithm can achieve the optimal payload distribution of multiple images steganography to improve the anti-statistical detection performance of stego images.

Keywords: Multiple images steganography; payloads distribution; dynamic programming; messages recovery

1 Introduction

Steganography is the hiding of secret messages in unsuspecting digital media, and the digital media can be transmitted in public channels, so that secret messages are not detected by third parties. The covers generally refer to publicly available digital content, including multimedia content such as images, video and audio. Currently, existing steganographic techniques [1–4] usually embed secret messages in a single image. If too much messages to be embedded, the stego images carrying more secret messages are very easy to be detected by a third party. In our daily work and

life, it is very common for users to upload and download images in batches on social networks. In order to transmit a large number of messages at one time, researchers have proposed some methods for multiple images steganography [5–7]. The sender separately embeds the messages into multiple cover images, and the batch of stego images is transmitted through the public channel. The receiver of the message can obtain multiple stego images at one time. When applying multiple steganography to the real-world, a sender usually has multiple images and a long message, the problem faced by this sender becomes how to allocate messages among multiple images to be the least detectable, which is the main research issue of multiple images steganography.

In the study of the multiple images steganography, Gasarch [8] first proposed the definition of the multiple images steganography model, and constructed a theoretical model based on the multiple images steganography technology. In the multiple images steganography model, the set of secret messages is embedded into multiple cover images through different steganographic techniques. After the transmission, the messages are extracted from multiple stego images and integrated. Although the preprocessing and embedding of messages are proposed in the multiple images steganography model, the problem of reasonable payload distribution when embedding messages is not considered. Assume that payload distribution is performed according to the features of the cover images, and messages are embedded in the cover images according to the size of the payload distribution. When the features of the stego images are close to the features of the cover images, the stego images are more likely to avoid detection by a third-party during transmission.

Aiming at the problem of reasonable payload distribution in multiple images steganography, Ker [9] first proposed the concepts of 'batch information hiding' and 'batch information analysis.' Ker [9] proposed the concept of 'batch information hiding' and believed that steganography should always focus the payloads on as few images as possible, or spread the payloads as sparsely as possible. However, the experimental results show that some batch detection and analysis methods can determine this payload distribution strategy based on experience, thereby they can effectively identify the cover images and the stego images. Ker et al. [10] studied the general batch steganalysis method and considered testing different payload distribution methods for the steganalysis method. This paper proposes five basic payload distribution algorithms and tests the multiple images steganography using actual data. The experimental results suggest that the payloads should be concentrated in as few cover images as possible or evenly distributed to each cover image. Due to the lack of set steganography analyzers, Ker et al. [11] proposed a blind universal pool steganography analyzer to verify the theoretical results of the multi-carrier steganography technology. The above methods study how to more reasonably distribute the payloads, but do not take into account the diversity of cover images.

Since the features and texture complexity of each cover image are different, the distortion values caused by embedding the same payload are also different. Considering the diversity of cover images in multiple images steganography, Zhao et al. [12,13], Cogranne et al. [14], Yu et al. [15] and Li et al. [16] proposed the latest development direction of multiple images steganography research. Zhao et al. [12] proposed an adaptive multiple images steganography method for spatial domain images based on distortion and steganography security. On this basis, Zhao et al. [13] proposes a general adaptive multiple images steganography method for spatial and JPEG domain images, and re-determines the amount of payloads embedded in each batch of images according to the "size" rule of the image and the histogram equalization to improve the detection error rate of stego images. Cogranne et al. [14] studied the practical strategy of distributing payloads in images, and distributed the payloads in all images, and using the statistical model as the output

of the detector, which further improved the steganography of multiple images anti-detection performance. Yu et al. [15] proposed to use the max-residual-greedy to embed secret messages, and use three methods to calculate the residual value of the cover image to select the cover image with the largest residual value for embedding. Experimental results show that the above methods embed secret messages into multiple images through different payload distribution algorithms, which effectively reduces the accuracy of the detection of the stego images.

In order to make the payloads more accurately distributed to multiple images, Li et al. [16] researched and developed a non-uniform payload distribution algorithm, and replaced the non-uniform payload distribution of multiple images by FBR (Feature Backward Replacement) algorithm to improve the anti-detection performance of stego images. At the same time, Li et al. [16] utilizes the data decomposition mechanism based on the Vandermonde matrix [17] to improve the embedding rate of the cover images and the robustness of secret messages. Even if a small part of the stego images are lost during transmission, the method can still extract messages from the remaining stego images. The algorithm improves the accuracy of each image payload distribution. After the FBR algorithm combining with different steganography algorithm for embedding. The detection accuracy rate of stego images is lower than that without FBR algorithm. However, although the algorithm takes into account the difference of each cover image during the payload distribution, it takes too many iterations during the payload distribution process. Therefore, when using this algorithm to embed multiple images, the required payload distribution time is longer.

The methods in Zhao et al. [12,13], Cogranne et al. [14] and Yu et al. [15] roughly and indirectly measures the complexity of the images, Li et al. [16] distributes payloads to multiple cover images more accurately at the expense of distribution efficiency. This paper considers the issues of accuracy and efficiency of payload distribution, the maximum embedding capacity of each image are determined by the features of the cover images. Then based on the idea of dynamic programming, the difference between the features of the cover images and the stego images is used to design the dynamic payload redistribution (DPR) algorithm. The DPR algorithm is expected to improve the embedding efficiency of secret messages and the fault tolerance during payload distribution. The improved DPR algorithm can be combined with traditional steganography such as LSB [18], nsF5 [2]. It can also be combined with the adaptive steganography algorithm such as J-UNIWARD [19] to improve the payload distribution efficiency of existing multiple images steganography and the detection error rate of stego images.

The structure of this paper is as follows: Section 2 mainly introduces the related work. Section 3 describes the main framework and steps of this method in detail, and gives the algorithm of payload distribution. Section 4 gives the experimental results and comparison of the proposed algorithm with existing algorithms. Finally, the paper is concluded in Section 5.

2 Related Works

For multiple images steganography methods, the value of the Maximum Mean Discrepancy is usually used to measure the difference between the features of cover sets and the features of stego sets. The smaller the value of MMD, the smaller the difference between the features of the image sets, and the less likely the stego images will be detected during transmission. This section mainly introduces the principle of Maximum mean discrepancy.

Maximum Mean Discrepancy (MMD) was proposed in [20] and used for double-sample detection to determine whether the two distributions p and q were the same. Pevny [21] proposed and proved that the value of MMD can be used to evaluate the difference between the cover set and the stego set. The features of the cover set and the stego set are regarded as two

different distributions. The MMD value represents the magnitude of the difference between the feature sets.

The statistical detection method based on MMD refers to: finding the continuous function f in the sample space based on two distributed samples, and finding the average value of two distributions on this continuous function f. By comparing the two mean values, the average difference between the two distributions corresponding to the function f can be obtained. Finding the function f that maximizes the average difference between the two distributions gives the value of MMD. Finally, the value of MMD is used to determine whether the two distributions are identical. If the value is small enough, the two distributions are considered identical, otherwise the two distributions are considered different.

MMD has been proved in [20] to distinguish the difference between the features of the cover images and the stego images. In practical steganography applications, assuming X is a separable measurement space, p_c and p_s are probability distributions defined on X, limiting the set of functions to a smaller sample range F, and the difference between p and q is measured as shown in Eq. (1).

$$MMD[F,p,q] = \sup_{f \in F}(E_{x\sim p_c} f(x) - E_{y\sim p_s} f(y)) \tag{1}$$

Suppose $X = \{x_1, \ldots, x_m\}$ is a dataset sample obtained from distributions p_c, the size of the data set is m, and $Y = \{y_1, \ldots, y_n\}$ is a dataset sample obtained from distributions p_s, the size is n. The two continuous functions distributed on the sample space are f, and the average values of the two values distributed on the continuous function are $E_{x\sim p_c} f(x)$ and $E_{y\sim p_s} f(y)$, respectively. The sup represents an upper bound in a set, that is, the smallest element greater than or equal to all other elements in the set, which is not necessarily in the set. Experience from MMD based on X and Y can be estimated as shown in Eq. (2).

$$MMD[F,X,Y] = \sup_{f \in F}\left(\frac{1}{m}\sum_{i=1}^{m} f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(y_i)\right) \tag{2}$$

It is proved in [15] that F is selected as a unit ball in Reproducing Kernel Hilbert Space (RKHS), then when $p_c = p_s$ exists, $MMD[F,p_c,p_s] = 0$; if $p_c \neq p_s$, $MMD[F,p_c,p_s] > 0$. Given further the kernel function K corresponding to RKHS, the square of this MMD can be expressed as shown in Eq. (3).

$$MMD^2[F,p,q] = E_{x,x'}\left[k\left(x,x'\right)\right] - 2E_{x,y}\left[k\left(x,y\right)\right] + E_{y,y'}\left[k\left(y,y'\right)\right] \tag{3}$$

where x and x' respectively are two subordinate random variables to p, y and y' are random variables to q. Estimate statistics for this MMD can be expressed as shown in Eq. (4).

$$MMD[F,X,Y] = \left[\frac{1}{m^2}\sum_{i,j=1}^{m} k\left(x_i,x_j\right) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k\left(x_i,y_i\right) + \frac{1}{n^2}\sum_{i,j=1}^{n} k\left(y_i.y_j\right)\right]^{\frac{1}{2}} \tag{4}$$

According to the above formula, the principle of MMD is to project and sum each sample, and use the size of the sum to measure the difference between the two sample distributions. In fact, the value of MMD is not only used to demonstrate the difference between the two features, but also to evaluate the security of the steganography algorithm [21]. Suppose that in RKHS, the

greater the value of MMD, the farther the distribution of the stego images and the cover images; the smaller the value of MMD, the closer the distribution of the stego images and the cover images. That is, when the distribution of the cover images in RKHS is unchanged, the smaller the MMD value, the closer the distribution of the cover images is to that of the stego images, the less easily the stego images will be detected.

Since the value of MMD can be used to evaluate the security of the steganography algorithm, for multiple images steganography, the value of MMD can be used to determine the optimal payload distribution for each cover image. In this paper, when designing the dynamic payload redistribution algorithm, the payloads are unevenly distributed to cover images by using the difference between the features of the cover images and the stego images, thereby reducing the value of MMD between the cover images and the stego images.

3 Proposed Method

Based on the idea of dynamic programming, this section proposes a dynamic payload redistribution (DPR) algorithm that uses the value of MMD to evaluate the difference between the features of the cover images and the stego images, to achieve the optimal payload distribution of the cover images, and to solve the problems of low embedding efficiency and low fault tolerance in existing payload distribution algorithms. When the DPR algorithm proposed in this section combined with the adaptive steganography algorithm, it is having better anti-statistical detection performance. Section 3.1 presents the main multiple images steganography method framework; Section 3.2 introduces the method of messages data decomposition and recovery; Section 3.3 proposes algorithms for payloads redistribution; Section 3.4 describes the process of multiple images embedding and extracting.

3.1 Main Framework

3.1.1 Framework Figure

Based on the requirements for image steganography in practical applications, this section proposes a framework for batch steganography of multiple images. The process of image steganography mainly has three parts: constructing the stego images, sending secret messages, and extracting secret messages. The multiple images steganography framework proposed in this paper is shown in Fig. 1. The importance of this method is mainly in the construction of the stego images. Firstly, the secret messages are decomposed by using a data decomposition mechanism based on the Vandermonde matrix, and then the payloads of the cover images are redistributed using a dynamic payload distribution algorithm. The stego images send messages through the public channel, and finally the secret messages are extracted and restored.

① Secret messages preprocessing: According to the number of cover images in multiple images steganography, the secret messages that need to be embedded are decomposed, and the secret messages are decomposed into different information blocks by the data decomposition mechanism based on the Vandermonde matrix. As shown in Fig. 1①, the decomposed secret information blocks can be directly assigned to the cover images.

② Dynamic payloads distribution: First, the decomposed messages are equally distributed to each cover image. After generating the stego images, extract the features of the cover images and the stego images separately. Based on the difference between the features of the cover images and the stego images, a dynamic payload redistribution algorithm is used to distribute the payloads of the cover images. As shown in Fig. 1②, each cover gets a new payload after dynamic payload redistribution algorithm.

③ Secret messages embedding and extracting: The secret messages are embedded in the cover images with the redistributed payloads. The constructed stego images are transmitted to the receiver through the channel transmission. During the transmission process, a small part of the images data will inevitably be lost or damaged. According to the characteristics of the Vandermonde matrix, as long as the receiver can receive sufficient number of secret images, part of the messages extracted from each secret image will restore the original secret messages, as shown in Fig. 1③.

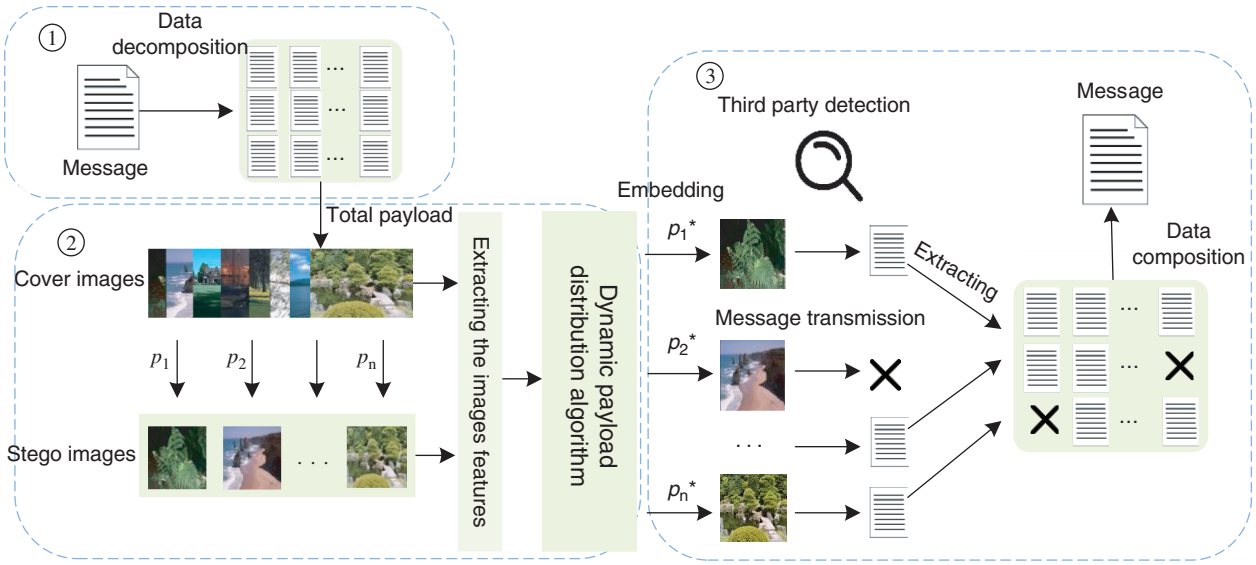

Figure 1: Multiple images steganography framework

3.1.2 Analysis

As can be seen from the frame diagram, the process of preprocessing the secret message is to decompose the secret message using a data decomposition mechanism based on the Vandermonde matrix. The combination of the secret message and the Vandermonde matrix produces data redundancy. The decomposed secret message becomes many different information blocks, and these information blocks are embedded in different cover images. Even if the messages are partially lost during transmission, as long as the data loss rate is less than the data redundancy rate, the receiver can use the completed information blocks to recover the original messages.

In the dynamic payload redistribution algorithm proposed in this paper, the value of the difference between the features of the cover set and the stego set is used to evaluate the security of the stego message. The difference between the cover set and the stego set is represented by the value of MMD. The smaller the value of MMD, the more similar the features of the cover set and the stego set, so the stego images are not easy to be detected. Conversely, if the value of MMD is larger, the stego images are more easily detected. The dynamic payload redistribution algorithm uses the principle of dynamic programming to distribute the payload non-uniformly in each cover image, so that the difference between the features of the cover set and the stego set are minimized. On the one hand, the process of dynamic programming shortens the time required for payloads distribution, and at the same time reduces the detection accuracy of stego images.

3.2 Decompose and Recovery the Secret Messages

In multiple images steganography, the steganography hides the secret messages in multiple cover images, and the receiver needs to receive all the stego images to fully recover the original messages. However, in the actual channel transmission process, it is difficult to achieve the complete transmission of the images, such as the influence of noise or the loss of some images content during the transmission process. In this section, we combine the secret messages with the Vandermonde matrix. The messages are divided into blocks and embedded in different cover images. In this way, even if part of the stego images are lost during transmission, the embedded information can still be extracted from the remaining part of the stego images.

We treat many given messages as binary streams, and convert the binary secret messages into a b-ary digit sequences, where b is an odd prime. First, the messages sequences are divided into K parts, each part contains L_1 digits. Convert each part to a b-ary sequence containing L_2 digits. The relationship between L_1 and L_2 is shown in Eq. (5).

$$L_1 = \lfloor L_2 \cdot \log_2 b \rfloor \tag{5}$$

For example, the messages are the binary sequence (1001 1101 0110). We divide the binary sequence into 3 parts, each part contains 4 digits, which is $L_1 = 4$. Assuming that $b = 5$, the binary sequence will be converted to (11 14 23). At this point, each part of the 5-ary sequence contains 2 digits, which is $L_2 = 2$.

According to Eq. (5), the redundancy rate Re can be represented as shown in Eq. (6).

$$\mathrm{Re} = 1 - \frac{L1}{L2 \cdot \log_2 b} < \frac{1}{L_1 + 1} \tag{6}$$

when L_1 and L_2 are large enough, the redundancy rate Re approaches 0. Therefore, we can directly treat binary messages sequence as b-ary sequence. The secret messages are transformed into K parts after conversion, each part contains L_2 digits, and each part can be expressed as shown in Eq. (7).

$$\left\{ d_{k,1} \quad d_{k,2} \quad \ldots \quad d_{k,L_1} \right\}, \quad k \in [1, K] \tag{7}$$

Build Vandermonde matrix A, and modulo q to matrix A, as shown in Eq. (8). The indices in matrix A are integers between 0 and $b-1$, which is $a_1, a_2, \ldots, a_{L_2} \in [0, b-1]$.

$$A = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ a_1 & a_2 & a_3 & \cdots & a_{L_2} \\ a_1^2 & a_2^2 & a_3^2 & \cdots & a_{L_2}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1^{L_1-1} & a_2^{L_1-1} & a_3^{L_1-1} & \cdots & a_{L_2}^{L_1-1} \end{bmatrix} \bmod b \tag{8}$$

where $L_1 \le L_2 \le b$ and $a_1 \ne a_2 \ne \cdots \ne a_{L_2}$ are satisfied in matrix A.

According to Li et al. [16], each messages are decomposed into n blocks, and the secret messages $\left\{ d_{k,1} \quad d_{k,2} \quad \ldots \quad d_{k,L_1} \right\}$ can be converted into $\left[s_{k,1} \quad s_{k,2} \quad \ldots \quad s_{k,L_2} \right]$ after being decomposed according to Eq. (9).

$$\left[s_{k,1} \quad s_{k,2} \quad \ldots \quad s_{k,L_2} \right] = \left[d_{k,1} \quad d_{k,2} \quad \ldots \quad d_{k,L_1} \right] \cdot A \tag{9}$$

where $\left[s_{k,1} \quad s_{k,2} \quad \ldots \quad s_{k,L_2} \right]$ corresponds $\left[a_1, a_2, \ldots, a_{L_2} \right]$ in matrix A.

For example, the messages data block is {2, 4, 1}, assuming $L_1 = 3$, $L_2 = 4$, $b = 5$, the randomly established Vandermonde matrix A is shown in Eq. (10).

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 4 & 0 & 1 \\ 4 & 1 & 0 & 1 \end{bmatrix} \tag{10}$$

According to Eq. (9), the messages data can be decomposed into: 4, 4, 2, 2.

The messages are embedded in each cover image after being decomposed. In the process of recovering the messages, it can be known from Eqs. (8) and (9) that the messages data of L_1 bits are decomposed and expanded into L_2-digit messages data for embedding, and it can be seen that the redundancy rate Re can be expressed as shown in Eq. (11).

$$Re = \frac{L_2 - L_1}{L_2} \tag{11}$$

The relationship between L_1 and L_2 needs to satisfy $L_1 \leq L_2$. If the data loss rate during transmission is greater than Re, the original messages data cannot be recovered; otherwise, we can recover the original messages data with the data ratio of L_1/L_2. It is assumed that all L_2 data blocks $\begin{bmatrix} s'_{k,1} & s'_{k,2} & \cdots & s'_{k,L_2} \end{bmatrix}$ are received at the data receiving end, which respectively correspond to $a'_1, a'_2, \ldots, a'_{L_2}$ in matrix A. According to Eq. (8) and the corresponding $a'_1, a'_2, \ldots, a'_{L_2}$ in the matrix, a Vandermonde matrix A' of size $L_2 \times L_2$ is established. The original data $\begin{bmatrix} d_{k,1} & d_{k,2} & \ldots & d_{k,L_2} \end{bmatrix}$ can be recovered by the following Eq. (12).

$$\begin{bmatrix} d_{k,1} & d_{k,2} & \ldots & d_{k,L_1} \end{bmatrix} = \begin{bmatrix} s'_{k,1} & s'_{k,2} & \cdots & s'_{k,L_2} \end{bmatrix} \cdot \left(A'\right)^{-1} \tag{12}$$

where $\left(A'\right)^{-1}$ is the inverse of matrix A'.

According to this method, a Vandermonde matrix is constructed to decompose the secret messages. Each part of the decomposed messages $\begin{bmatrix} s_{k,1} & s_{k,2} & \ldots & s_{k,L_2} \end{bmatrix}$ corresponds to the indices $\begin{bmatrix} a_1, a_2, \ldots, a_{L_2} \end{bmatrix}$ of the Vandermonde matrix, and the decomposed messages are embedded into each cover image together with the indices of the Vandermonde matrix through different steganography algorithms. After the transmission of the channel, the messages $\begin{bmatrix} s'_{k,1} & s'_{k,2} & \cdots & s'_{k,L_2} \end{bmatrix}$ and the corresponding indices in the stego images are extracted, and the $a'_1, a'_2, \ldots, a'_{L_2}$ recovers the original messages from the extracted messages.

3.3 Dynamic Payload Redistribution Algorithm

The traditional multiple images steganography algorithm distributes secret messages evenly in each image. This equal distribution scheme does not take into account the features of each cover image, making the stego images easy to detect. In the method of this paper, in order to improve the security of message transmission, according to the diversity of the cover images, and the idea of dynamic programming is used to distribute the payload non-uniformly to each image to obtain the optimal payload distribution method.

Assume that given N cover images $(\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_N)$, the length of the messages is P. In the traditional multiple images steganography method, the messages P are evenly distributed among

N cover images, and the same length of the secret information embedded in each cover image is shown as Eq. (13).

$$p_1 = p_2 = p_3 = \ldots = p_N = \frac{P}{N} \tag{13}$$

Since each cover image is different, the method of evenly embedding secret messages in the cover image makes it easier to detect the stego images. This paper proposes an optimization algorithm for payload redistribution. According to the principle of minimizing the difference between the feature of the cover images and the stego images, the payloads size of each cover image are redistributed to improve the security of multiple images steganography.

In the multiple images steganography method, the value of MMD is usually used to evaluate the difference between the features of the cover images and the stego images. The smaller the value of MMD between the cover images and the stego images, the closer the features of the cover images and the stego images. The stego images are even more difficult to detect.

Assume that the non-uniform payloads of the cover images after iteration are $\left(p_1^*, p_2^*, \ldots, p_N^*\right)$, and the constraints are satisfied between each payload as shown in Eq. (14).

$$p_1^* + p_2^* + \ldots + p_N^* = P \tag{14}$$

When the constraint condition Eq. (14) is satisfied, the value of MMD is used as the criterion for evaluating safety. These non-uniform payloads distribution are expressed as shown in Eq. (15).

$$\left(p_1^*, p_2^*, \ldots, p_N^*\right) = \underset{F_C \in \Re, F_S \in \Re, F_{S^*} \in \Re}{\operatorname{argmin}} \left\{\mathrm{MMD}\left(\mathrm{F_C}, \mathrm{F_S}\right), \mathrm{MMD}\left(\mathrm{F_C}, \mathrm{F_{S^*}}\right)\right\} \tag{15}$$

where F_C is the feature set of the cover images $(\mathrm{C}_1, \mathrm{C}_2, \ldots, C_N)$, F_S is the feature set of the stego images $(\mathrm{S}_1, \mathrm{S}_2, \ldots \mathrm{S}_N)$ after the secret messages are evenly embedded by the traditional steganography method, and F_{S^*} is the stego images feature set that re-embeds messages after payloads redistribution. All sets belong to the real number $\Re$.

The redistributed payloads $\left(p_1^*, p_2^*, \ldots, p_N^*\right)$ are used to embed the messages to ensure that the obtained MMD value is the smallest and improve the security of the multiple images steganography. The DPR algorithm proposed in this paper uses the principle of dynamic programming. By reducing the number of iterations during payload distribution, the computational complexity of batch processing images is reduced, and the efficiency of the algorithm is improved. Compared with the FBR algorithm proposed in Eq. (16), the algorithm in this paper is expected to improve the time complexity and fault tolerance. The DPR algorithm is as follows:

Algorithm 1: Dynamic payload redistribution algorithm

Input: Cover images $\{C_1, \mathrm{C}_2, \ldots, \mathrm{C}_N\}$, Messages D, Average payloads $(p_1, p_2, \ldots, p_N)$;

Output: Redistributed payloads $\left(p_1^*, p_2^*, \ldots, p_N^*\right)$, Stego images $\{S_1, S_2, \ldots, S_N\}$.

$for(i = 1, i <= N, i++)$

Step 1: Embedding secret messages.

Secret messages D are embedded into the cover images at the average payloads of $(p_1, p_2, \ldots, p_N)$ to generate the stego images. The PEV-274 dimension features of the cover

(Continued)

Algorithm 1 (Continued)

images and the stego images are extracted using the feature extraction method in [21]. The cover images feature F_C and the stego images feature F_S are extracted.

Step 2: Calculate the MMD value of the features of the cover images and the stego images.

Based on the image features obtained in Step 1, when the payloads are evenly distributed, the value of the features difference MMD (F_C, F_S) between the cover images and the stego images are recorded as $M1$.

Step 3: Dynamic payload redistribution.

According to the embedding rate of each cover image, the maximum value of the embeddable information blocks is determined. After embedding, the value of MMD between image features are calculated and recorded as $M2$. $M2$ is compared with $M1$ in Step 2. If $M1 > M2$, the value of $M2$ is retained; if $M1 < M2$, the number of information blocks are sequentially reduced and embedded, until $M1 > M2$, the payloads distribution result at this time are retained. The number of information blocks embedded in each image is recorded as $p(i)$, and the corresponding MMD value is recorded as $m(i)$. Using the idea of dynamic programming to redistribute the payloads. Assuming that in the i-th image, the payload is $p(i)$, then the state transition equation is shown in Eq. (16).

$$f[i][j] = \arg\min\{f[i-1][j-p(i)] + m(i), f[\mathrm{i}-1][j]\} \tag{16}$$

After the i-th image embeds the payloads j, $f[i][j]$ is the smallest MMD value of the cover images and the stego images. Compare the MMD values of the i-th image before and after embedding, and select the minimum value to assign to $f[i][j]$. Specific examples of the planning process are shown in Tab. 1.

Step 4: Determine the redistributed payloads.

Step 3 recursively obtain the payloads $(p_1^*, p_2^*, \ldots, p_N^*)$ of the cover images after payload redistribution, and calculate the total payload $\mathrm{P}^* = p_1^* + p_2^* + \ldots + p_N^*$. If $P^* \geq P$, embed directly according to the redistributed payloads; if $P^* < P$, all the extra payload $p^* = P - P^*$ is embedded in the last cover image to ensure $p_1^* + p_2^* + \ldots + p_N^* = \mathrm{P}$.

Step 5: Output the redistributed payloads $(p_1^*, p_2^*, \ldots, p_N^*)$.

end.

An example of the planning process in Step 3 is as follows. Assume that there are four images with the numbers A, B, C, and D. The embedded information blocks p of each image can be 2, 1, 4, 3, and the MMD values corresponding to the stego images after embedding are respectively 0.4, 0.3, 0.1, 0.3, as shown in Tab. 1. Now the total number of payloads of the embedded secret messages is 7. Using the above dynamic programming method, after embedding the total payload, the values of MMD corresponding to the cover images and the stego images are the smallest.

Assume that picture A is used for embedding, and the state transition equation at this time is $f[4][7] = \arg\min\{f[3][7-p(a)] + m(a), f[3][7]\}$, where $f[3][7-p(a)] + m(a)$ represents the value of MMD when using picture A for embedding, and $f[3][7]$ represents the value of MMD when

not using picture A for embedding. Compare the size of the MMD value in the two cases, and choose a smaller value to assign to f [4][7]. According to the payload planning process shown in Tab. 1, when the total payload is 7, there are two alternative embedding schemes. The first solution is to use the A, B, and C images for embedding. The second solution is to use the C and D images for embedding. Compare the MMD values of the two solutions, and choose the one with the smaller MMD value. The secret messages are embedded in the two cover images of C and D, and the MMD value obtained at this time is the smallest.

Table 1: Payload planning process

	p	m	0	1	2	3	4	5	6	7
A	2	0.4	0	0	0.4A	–	–	–	–	–
B	1	0.3	0	0.3B	0.4A	0.7AB	–	–	–	–
C	4	0.1	0	0.3B	0.4A	0.7AB	0.1C	0.4BC	0.5AC	0.8ABC
D	3	0.3	0	0.3B	0.4A	0.3D	0.1C	0.4BC	0.5AC	0.4CD

3.4 Embedding and Extracting Secret Messages

This section combines the traditional steganography with the DPR algorithm, and uses the data decomposition method of Section 3.2 to decompose the secret message to be embedded. The decomposed messages are embedded into the cover images using the DPR algorithm of Section 3.3. The messages embedding algorithm is as follows:

Algorithm 2: Secret messages embedding Algorithm

Input: Cover images, Secret messages

Output: Stego images

Step 1: Decomposition of secret messages.

According to the messages decomposition method proposed in Section 3.2, the embedded messages are decomposed, and the Vandermonde matrix is constructed to decompose the original messages into multiple information blocks, corresponding to the indices $[a_1, a_2, \ldots, a_{L_2}]$ in the Vandermonde matrix.

Step 2: Redistribute the payloads.

The size of the total payload to be embedded is determined by the decomposed messages, and the total payload P is non-uniformly distributed to each cover images according to the DPR algorithm proposed in Section 3.3.

Step 3: Embedding of secret messages.

According to the redistributed payloads $(p_1^*, p_2^*, \ldots, p_N^*)$, the cover images are embedded in batches using the steganography algorithm.

Combined with the data recovery method proposed in Section 3.2, the secret messages in the stego images are extracted. The secret messages extraction algorithm is as follows:

Algorithm 3: Secret messages extraction algorithm

Input: Stego images

Output: Cover images

Step 1: Receive stego images.

The transmission of the stego images through the channel may cause the loss of some images, and the information blocks in the received stego images are extracted, as well as the indices $[a_1, a_2, \ldots, a_{L_2}]$ in the Vandermonde matrix corresponding to the information blocks.

Step 2: Extraction of messages.

According to the data recovery method proposed in Section 3.2. The $L_2 \times L_2$ Vandermonde matrix A' is constructed by the indices $[a_1, a_2, \ldots, a_{L_2}]$ corresponding to the messages blocks.

Step 3: Recovery of the original secret messages.

Calculate the original secret messages from Eq. (12).

According to the secret data recovery method, it can be known that the data redundancy rate Re is controlled by two parameters L_1 and L_2 of the Vandermonde matrix, and the maximum amount of data allowed to be lost during transmission is $L_1 - L_2$. If too much data are lost during transmission, the original data cannot be recovered using the data recovery method proposed in this paper.

4 Experimental Results

In this section, to verify the performance of the proposed method, several experiments are conducted in this section comparing with previous representative methods, a series of experiments were performed in terms of payload distribution efficiency and anti-statistical detection compared with existing multiple images steganography methods. Finally, the robustness of the data decomposition model based on Vandermonde matrix is verified.

4.1 Performance Comparisons

To verify the performance of the proposed method, an experimental comparison is made with the existing multiple images steganography methods. First, the experimental settings are given in the following section. Then, the experimental results are illustrated and analyzed in the terms of the quality of stego images, the efficiency of payload distribution under different multiple images steganography methods, and the anti-detection performance.

4.1.1 Experimental Settings

To verify the applicability of proposed method for image types, cover images come from three image databases are tested: (1) Bossbase-1.01 image database [22]; (2) UCID image database [23]; (3) GTD image database [24]. Randomly select 2000 images from the above three image databases. These images by JPEG2000 lossless compression to generate 2000 JPEG images. The images size are 512×512, and the quality factor are 85. The cover images use different multiple images steganography algorithms in combination with the LSB steganography [18], nsF5 steganography [2], and J-UNIWARD steganography [19]. Randomly select images from the cover images to

generate stego images under different payloads. Compare different multiple images steganography algorithms FBR [16], ES-ITC [25] and EVEN [10]. The experimental settings are shown in Tab. 2.

Table 2: Experimental settings

Experimental settings	
Image sources	Bossbase-1.01 [22], UCID [23], GTD [24]
Image size	512×512
Image type	JPEG
Number of covers	2000
Secret messages	Randomly generated binary sequences
Data redundancy rate	44.44%, 83.87%
Payloads	0.05, 0.2, 0.4
Steganography	nsF5 [2], LSB [18], J-UNIWARD [19]
Distribution algorithms	FBR [16], ES-ITC [25], EVEN [10]
Proposed algorithm	DPR
Detection feature	DCTR [26]

During the experiment, we can choose to include different numbers of cover images in each group of images, and use different embedding algorithms nsF5 [2], LSB [18], J-UNIWARD [19] to experiment with different numbers of cover images. Assuming that each group of images contains 50 cover images, the average payloads are 0.05 bpp, 0.2 bpp and 0.4 bpp. The detailed experimental steps are as follows:

① All cover images are grouped according to each group containing 50 images, and messages are distributed according to the number of images and embedded in each cover image.

② The feature extraction method of PEV-274 is used to extract the features of the cover images set and the stego images set, and the MMD value are calculated based on the differences between the features of cover images and the stego images.

③ According to the dynamic payload redistribution algorithm, the idea of dynamic programming is used to redistribute the payloads of the cover images.

④ According to the payloads of the redistributed cover images, combined with different embedding algorithms to embed messages to get the stego images.

4.1.2 Stego Images Quality Comparison

This section compares the DPR algorithm proposed in this paper with the FBR algorithm in [16] by the PSNR value of the stego images. The cover images set was randomly divided into 40 groups, each group containing 50 cover images. A group of images is randomly selected from 40 groups of cover images, and the LSB steganography [18] and the nsF5 steganography [2] are combined with the DPR algorithm and the FBR algorithm [16], respectively. The PSNR values of the generated stego images are compared at different payloads. The comparison of the PSNR values of the stego images generated by combining LSB steganography [18] with DPR algorithm and FBR algorithm [16] is shown in Tab. 3. As can be seen from Tab. 3, comparing the PSNR value of the stego images generated by the DPR algorithm and the FBR algorithm [16], the PSNR value can be increased by a maximum of 1.89 when the average payloads is 0.1 bpp. The comparison of the PSNR values of the stego images when using nsF5 steganography [2] is shown

in Tab. 4, the PSNR value can be increased by a maximum of 1.3 when the average payloads is 0.1 bpp. As shown in Fig. 2, the PSNR value of the stego images generated by the DPR algorithm proposed in this paper under different payloads is equivalent to the PSNR of the stego images obtained by the FBR algorithm in [16].

Table 3: Comparison of PSNR values combined with LSB steganography

Algorithm	Payload							
	0.01	0.03	0.05	0.07	0.09	0.1	0.15	0.2
FBR [16]	53.59	50.09	48.22	45.97	44.42	43.84	43.01	42.5
Proposed method	53.62	49.76	47.91	46.67	45.65	45.73	43.43	43.16

Table 4: Comparison of PSNR values combined with nsF5 steganography

Algorithm	Payload							
	0.01	0.03	0.05	0.07	0.09	0.1	0.15	0.2
FBR [16]	73.54	71.22	67.39	65.59	62.73	61.69	58.81	56.39
Proposed method	73.91	68.90	66.38	64.63	63.35	62.99	59.51	56.26

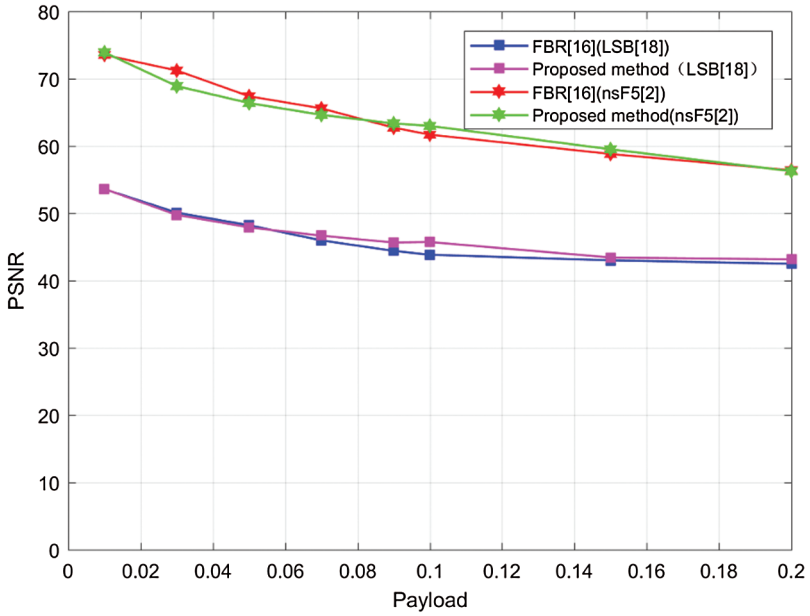

Figure 2: Comparison of PSNR values at different payloads

4.1.3 Payload Distribution Efficiency Comparison

This section compares the DPR algorithm proposed in this paper, FBR algorithm [16] and ES-ITC algorithm [25] from the time required for payloads distribution. Taking the nsF5 algorithm [2] as an example, a set of cover images is randomly selected from the 40 sets of

cover images, combined with different multiple images steganography methods, and embedded at the payloads of 0.05, 0.2, 0.4. This experiment was repeated 10 times to calculate the average time required for payloads distribution. Compare the time required for payloads distribution with different multiple images steganography methods. At different payloads, when the nsF5 steganography [2] is used in combination with different multiple images steganography methods, the time required to distribute the payloads is shown in Tab. 5. As can be seen from Tab. 5, compared with the ES-ITC algorithm [25], the time required to distribute the payloads are increased by 0.2 s, 0.12 s and 0.13 s when the average payloads are 0.05 bpp, 0.2 bpp and 0.4 bpp. Compared with the FBR algorithm [15], the time required to distribute the payloads of the DPR algorithm are increased by 15.01 s, 14.29 s and 14.5 s when the average payloads are 0.05 bpp, 0.2 bpp and 0.4 bpp. It can be seen that when the nsF5 steganography [2] is combined with the DPR algorithm, the time required to distribute the payloads is generally less than the time required to distribute the payloads by the FBR algorithm [16] and ES-ITC algorithm [25]. Especially compared with the FBR algorithm, the time required for the DPR algorithm proposed in this paper to distribute the payloads can be reduced by 4 times.

Table 5: Comparison of payload distribution time between different distribution algorithms

Algorithm	Payload		
	0.05	0.2	0.4
ES-ITC [25]	5.11 s	5.14 s	5.07 s
FBR [16]	19.92 s	19.31 s	19.44 s
Proposed method	4.91 s	4.92 s	4.86 s

The FBR algorithm proposed in Eq. (16) obtains the non-uniform payloads distribution for each cover image by iterating the substitution sequence, but the efficiency of iteration substitution is not considered. Based on the Li et al. [16], this paper uses the principle of dynamic programming, and then redistributes the payloads of the cover images based on the difference between the features of the cover images and the stego images, so that the payload distribution efficiency has improved.

4.1.4 Anti-Statistical Detection Comparison

This section verifies the performance of the proposed method in terms of anti-statistical detection, using the nsF5 steganography [2] and the J-UNIWARD steganography [19], respectively, in combination with the DPR algorithm, the ES-ITC algorithm [25] and the EVEN algorithm [10]. All 40 groups of cover images were embedded with payloads of 0.05, 0.2, and 0.4. In the anti-statistical detection experiment, the 8000-dimensional DCTR feature [26] and ensemble classifier [27] were used to detect the anti-detection performance of the stego images generated by the method in this paper. The experimental cover images are randomly divided into two parts, namely the training cover images and the test cover images. The stego images generated by using different steganography algorithms and different payloads are correspondingly divided into training stego images and test stego images. The 8000-dimensional DCTR features are extracted from each training image set, and the steganography detector is trained using an ensemble classifier [27] using a supervised learning method. Finally, training classifiers are extracted from each training image set, and the minimum global average error rate with a prior

probability of equal probability is calculated using the resulting steganography detector as shown in (17).

$$P_E = \min_{P_{FA} \in [0,1]} \frac{1}{2} \left(P_{FA} + P_{MD} \left(P_{FA} \right) \right) \tag{17}$$

where P_{FA} is the false alarm rate, the probability that the cover images are determined to be the stego images, and P_{MD} is the missed detection rate,the probability that the stego images are to be the cover images. For each group of experiments, the above process was repeated 10 times, and the minimum median average error rate of 10 experiments is used as a measure of anti-statistical detection performance. The larger value indicates a better anti-statistical detection performance.

The Tab. 6 shows the average detection error rates of nsF5 steganography and J-UNIWARD steganography combined with different multiple images steganography methods, respectively. It can be seen from Fig. 3 that for the nsF5 steganography [2], the average detection error rate using the ES-ITC algorithm [25] is slightly better than the EVEN algorithm [10] and the DPR algorithm proposed in this paper; for the J-UNIWARD steganography [19], the average detection error rate using the DPR algorithm proposed in this paper is generally better than the other two algorithms. Compared with the EVEN algorithm [10] proposed in this paper, the average detection error rate increased by 0.6%, 0.42%, and 1.06% when the payloads were 0.05, 0.2, and 0.4; compared with the ES-ITC algorithm [25], the average detection error rate of the DPR algorithm increased by 0.17%, 0.15%, and 0.59% under different payloads, respectively.

Table 6: Comparison of average detection error rate of different distribution algorithms

Steganography	Algorithms	Payload		
		0.05	0.2	0.4
nsF5 [2]	EVEN [10]	0.3290	0.1324	0.0644
	ES-ITC [25]	0.3412	0.1590	0.0726
	Proposed method	0.3468	0.1489	0.0713
J-UNIWARD [19]	EVEN [10]	0.4510	0.3458	0.1540
	ES-ITC [25]	0.4527	0.3473	0.1587
	Proposed method	0.4570	0.3500	0.1646

The Even algorithm [10] distributes secret messages evenly to each cover image, and lacks consideration of the features of each cover image. The ES-ITC algorithm [25] is based on the texture complexity of the cover images and assigns the maximum capacity payload to each cover image. Based on the Li et al. [16], this paper uses the idea of dynamic programming to design the DPR algorithm. It can be seen that in multiple images steganography, the J-UNIWARD [19] adaptive steganography algorithm combined with the DPR algorithm proposed in this paper generates stego images with improved anti-statistical detection performance.

4.2 Robustness Experiments

In this paper, a data decomposition method based on the Vandermonde matrix is designed to improve the robustness of the messages in the batch steganography algorithm. The original secret messages are recovered according to the Eq. (11) in Section 3.2.

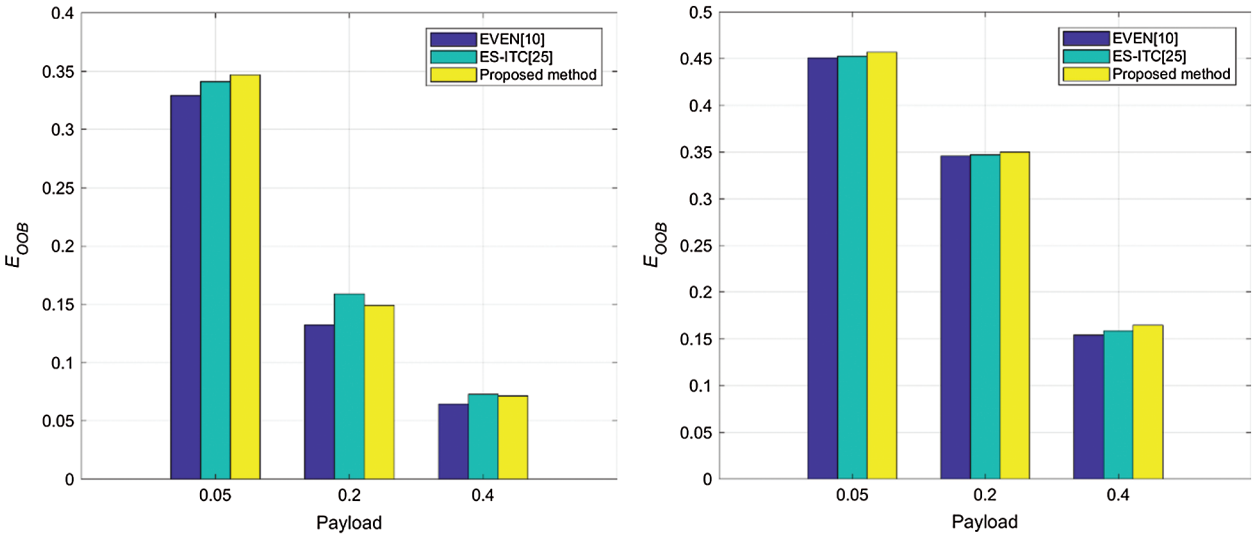

Figure 3: Comparison of average detection error rates of stego images with different allocation algorithms. (a) nsF5 steganography, (b) J-UNIWARD steganography

Assuming that the original messages data are the sequence of q-ary digits in L_1-bit, which is expanded to the sequence of q-ary digits in L_2-bit after data decomposition, the redundancy rate of the data is expressed as $Re = L_2 - L_1/L_2$ by Eq. (9). According to the data recovery process, as long as the receiver receives $L_2 - L_1$ bits in the L_2-bit messages, that is, when the data loss rate is greater than the data redundancy rate, the original messages can be recovered. Fig. 4 shows the relationship between data redundancy and parameters L_1 and L_2. For different parameter L_1, the data redundancy rate increases with the increase of parameter L_2.

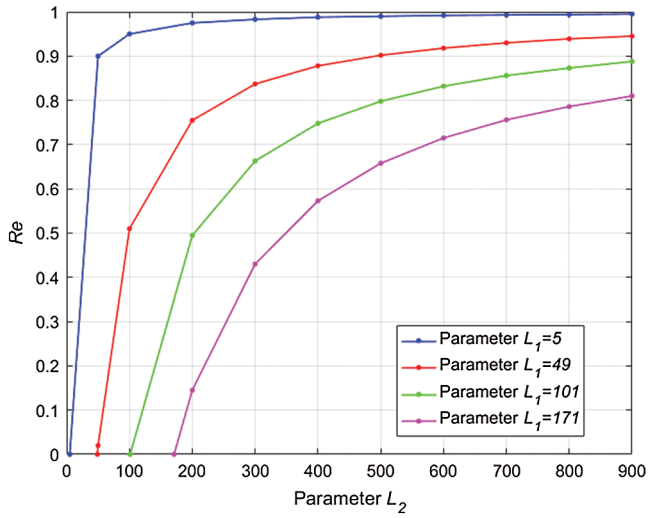

Figure 4: Relationship between Re and parameters L_1 and L_2

To verify the robustness of this method, the following experiments were performed. Using the secret messages extraction algorithm in Section 3.4, test three payloads of 0.025 bpp, 0.15 bpp, and 0.25 bpp, which respectively represent smaller payloads, normal payloads, and larger payloads. Two sets of test parameters are given: $L_1 = 5$, $L_2 = 9$, $q = 11$ and $L_1 = 5$, $L_2 = 31$, $q = 37$, the data redundancy rate Re of the two sets of parameters are 44.44% and 83.87%, respectively. The experiment was performed 100 times to simulate the loss of the stego images during transmission. The average data recovery rate was obtained by dividing the number of times the original data was recovered by the total number of tests. Tab. 7 shows the average data recovery rate of the three steganography algorithms combined with the secret message extraction method proposed in this paper when the data redundancy rate is 44.44% and the simulated average data loss rates are 30% and 50%. It can be seen from Tab. 7 that when the average data loss rate is about 30%, the data redundancy rate is greater than the average data loss rate, so in most cases secret messages can be recovered. When the average data loss rate is about 50%, the data redundancy rate is less than the average data loss rate. Too much lost data makes it impossible to recover the messages, so the average data recovery rate is 0%. Similarly, Tab. 8 shows the average data recovery rate of the three steganography algorithms combined with the secret message extraction method proposed in this paper when the data redundancy rate is 83.87% and the simulated average data loss rates are 80% and 90%. It can be seen that no matter which steganography algorithm is used, secret messages can be recovered as long as the average rate of data loss is less than the rate of data redundancy.

Table 7: Recovery rate of secret messages under different data loss rates ($Re = 44.44\%$)

Algorithm	Re	Average data loss	Payload		
			0.025	0.15	0.25
DPR-LSB	44.44%	31.24%	93%	85%	73%
		50.68%	0%	0%	0%
DPR-nsF5	44.44%	30.82%	88%	76%	69%
		51.44%	0%	0%	0%
DPR-J-UNIWARD	44.44%	33.49%	84%	72%	65%
		51.72%	0%	0%	0%

Table 8: Recovery rate of secret messages under different data loss rates ($Re = 83.87\%$)

Algorithm	Re	Average data loss	Payload		
			0.025	0.15	0.25
DPR-LSB	83.87%	79.28%	87%	79%	70%
		90.17%	0%	0%	0%
DPR-nsF5	83.87%	80.33%	78%	62%	54%
		89.43%	0%	0%	0%
DPR-J-UNIWARD	83.87%	81.27%	73%	56%	43%
		91.24%	0%	0%	0%

5 Conclusions

The current image steganography mainly focuses on the steganography of a single cover image. Existing algorithms for multiple images steganography often consider embedding secret messages as sparse as possible in the cover images to improve the detection error rate of the stego images, failing to take full advantage of the features of cover images in multiple images steganography. In order to distribute secret messages to each cover image more efficiently and accurately, this paper presents a DPR algorithm for JPEG images based on optimal payload distribution. Using the differences between the features of the cover images and the stego images and the idea of dynamic programming. The payloads are non-uniformly distributed to each cover image to achieve the optimal payloads distribution for multiple images steganography. The experimental results show that the proposed DPR algorithm combined with the steganography is more efficient than the FBR algorithm [16] and the ES-ITC algorithm [25] in distributing the payloads. Especially when compared with the FBR algorithm, the time required for payload distribution is reduced by 4 times. As the same time, the average detection error rates of J-UNIWARD [19] adaptive steganography combined with the DPR algorithm proposed in this paper is generally higher than that of EVEN algorithm [10] and ES-ITC algorithm [25]. When the embedding rate is 0.4 bpp, the average detection error rate of the DPR algorithm is 1.06% higher than the EVEN algorithm [10], and 0.59% higher than the ES-ITC algorithm [25].

In the next work, we will focus our research on different methods of selecting cover images in multiple steganography. The cover images most suitable for steganography in multiple covers are selected for embedding to improve the security of batch steganography.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Nos. U1736214, U1804263, U1636219, 61772281, 61772549, and 61872448), the National Key R&D Program of China (Nos. 2016YFB0801303, 2016QY01W0105) and the Science and Technology Innovation Talent Project of Henan Province (No. 184200510018).

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

References

1. Westfeld, A. (2001). F5-a steganographic algorithm-high capacity despite better steganalysis. *International Workshop on Information Hiding, 2137,* 289–302.
2. Fridrich, J., Pevný, T., Kodovský, J. (2007). Statistically undetectable JPEG steganography: dead ends challenges, and opportunities. *Workshop on Multimedia & Security, 2007,* 3–14.
3. Khan, S., Irfan, M. A., Arif, A., Tahir, S. T. H., Gul, A. et al. (2019). On hiding secret information in medium frequency DCT components using least significant bits steganography. *Computer Modeling in Engineering & Sciences, 118(3),* 529–546. DOI 10.31614/cmes.2019.06179.
4. Meng, R., Cui, Q., Yuan, C. (2018). A survey of image information hiding algorithms based on deep learning. *Computer Modeling in Engineering & Sciences, 117(3),* 425–454. DOI 10.31614/cmes.2018.04765.
5. Denemark, T., Fridrich, J. (2017). Steganography with multiple JPEG images of the same scene. *IEEE Transactions on Information Forensics and Security, 12(10),* 2308–2319. DOI 10.1109/TIFS.2017.2705625.
6. Ker, A. D. (2007). Batch steganography and pooled steganalysis. *Workshop on Information Hiding Workshop, 4437,* 265–281.
7. Pevný, T., Nikolaev, I. (2015). Optimizing pooling function for pooled steganalysis. *IEEE International Workshop on Information Forensics and Security, 2015,* 1–6.

8. Gasarch, W. I. (2001). Review of "The CodeBreakers: the story of secrete writing" by David Kahn. Scribner. *ACM SIGACT News, 32(2),* 5–6. DOI 10.1145/504192.1005762.
9. Ker, A. D. (2007). Batch Steganography and the threshold game. *Security, Steganography, and Watermarking of Multimedia Contents IX, 6505,* 0401–0413.
10. Ker, A. D., Pevny, T. (2012). Batch steganography in the real world. *Multimedia and Security-MM&Sec, 2012,* 1–10.
11. Ker, A. D., Pevny, T. (2012). Identifying a steganographer in realistic and heterogeneous data sets. *International Conference on Media Watermarking, Security, and Forensics,* 1–13.
12. Zhao, Z. Z., Guan, Q. X., Zhao, X. F., Yu, H. B., Liu, C. J. (2017). Embedding strategy for batch adaptive steganography. *International Workshop on Digital Watermarking, 10082,* 494–505.
13. Zhao, Z., Guan, Q., Zhao, X., Yu, H., Liu, C. (2018). Universal embedding strategy for batch adaptive steganography in both spatial and JPEG domain. *Multimedia Tools and Applications, 77(11),* 14093–14113. DOI 10.1007/s11042-017-5016-z.
14. Cogranne, R., Sedighi, V., Fridrich, J. (2017). Practical strategies for content-adaptive batch steganography and pooled steganalysis. *International Conference on Acoustics, Speech and Signal Processing, 2017,* 2122–2126.
15. Yu, X., Chen, K., Zhang, W., Wang, Y., Yu, N. (2019). Improving the embedding strategy for batch adaptive steganography. *Digital Forensics and Watermarking, 11378,* 248–260.
16. Li, F., Wu, K., Zhang, X., Yu, J., Lei, J. et al. (2018). Robust batch steganography in social networks with non-uniform payload and data decomposition. *IEEE Access, 6,* 29912–29925. DOI 10.1109/ACCESS.2018.2841415.
17. Zhang, X., Wang, S., Zhang, W. (2009). Steganography combining data decomposition mechanism and stego-coding method. *Informatica, 33(1),* 41–48.
18. Celik, M. U., Sharma, G., Tekalp, A. M., Saber, E. (2015). Lossless generalized-LSB data embedding. *IEEE Transactions on Image Processing, 14(2),* 253–266. DOI 10.1109/TIP.2004.840686.
19. Holub, V., Fridrich, J., Denemark, T. (2014). Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security, 2014(1),* 1–13. DOI 10.1186/1687-417X-2014-1.
20. Gretton, A., Borgwardt, M., Rasch, M., Schölkopf, B., Smola, A. J. (2007). A kernel method for the two-sample-problem. *Advances in Neural Information Processing Systems, 2007,* 513–520.
21. Pevny, T. (2008). *Kernel methods in steganalysis (Ph.D. Thesis)*. University of New York, Binghamton.
22. Bas, P., Filter, T., Pevny, T. (2011). Break our steganographic system: the ins and outs of organizing BOSS. *Information Hiding Conference, 96(454),* 488–499.
23. Gerald Schaefer, M. S. (2004). UCID: an uncompressed color image database. *International Society for Optical Engineering, 5307,* 472–480.
24. Barh, D., Kumar, A., Misra, A. N. (2010). Genomic target database (GTD): a database of potential targets in human pathogenic bacteria. *Bioinformation, 4(1),* 50–51. DOI 10.6026/97320630004050.
25. Liao, X., Yin, J. (2018). Two embedding strategies for payload distribution in multiple images steganography. *IEEE International Conference on Acoustics, Speech and Signal Processing,* 1982–1986. DOI 10.1109/ICASSP.2018.8462384.
26. Holub, V., Fridrich, J. (2015). Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Transactions on Information Forensics and Security, 10(2),* 219–228. DOI 10.1109/TIFS.2014.2364918.
27. Kodovský, J., Fridrich, J., Holub, V. (2012). Ensemble classifiers for steganalysis of digital media. *Information Forensics and Security, 7(2),* 432–444. DOI 10.1109/TIFS.2011.2175919.