applied sciences

MDPI

Article

Multi-Hider Reversible Data Hiding Using a Weighted Color Transfer and Modulus Operation

Ku-Sung Hsieh and Chung-Ming Wang *

Department of Computer Science and Engineering, National Chung Hsing University, Taichung 40227, Taiwan
* Correspondence: cmwang@cs.nchu.edu.tw

Abstract: This paper proposes a multi-hider dual-image reversible data-hiding algorithm. Generating dual stego images, the proposed algorithm provides a high embedding capacity, exhibits high image quality, offers reversibility to restore the stego images, and suggests seven levels of secret message extraction. The message embedding in our algorithm contains two phases. In Phase-1, we embed an *n*-ary secret message vector followed by an *m*-ary secret message in each pixel using two distinct secret keys when rendering a new image from the source and target images using an optimal weighted color-transfer process. This phase produces a tentative stego image which exhibits a new color appearance, different from the input source image. In Phase-2, we introduce a weighted modulus operation to embed a *k*-ary secret message vector into a dual-pixel constructed from the tentative stego image. The message concealment is elaborately designed so that the hidden secret messages are intact. This phase produces dual stego images which carry three distinct secret messages. Using legitimate secret keys, an authorized receiver can extract one or parts of secret messages and recover the tentative stego color-transferred image without arousing any suspicion. The experimental results and analysis confirm that our scheme can resist RS and PVD steganalytic attacks. In addition, our algorithm provides both high embedding capacity and better image quality, outperforming its counterparts. To the best of our knowledge, our scheme is the first in the relevant literature that provides multi-hider reversible data hiding, thereby offering various levels of message extraction for secure data communication.

Keywords: reversible data hiding; multi-hider message embedding; optimal weighted color transfer; weighted modulus embedding; RS steganalysis; security

Citation: Hsieh, K.-S.; Wang, C.-M. Multi-Hider Reversible Data Hiding Using a Weighted Color Transfer and Modulus Operation. *Appl. Sci.* **2023**, *13*, 1013. https://doi.org/10.3390/app13021013

Academic Editor: David Megías

Received: 12 November 2022
Revised: 5 January 2023
Accepted: 6 January 2023
Published: 11 January 2023

1. Introduction

The security of data communication plays a crucial role with the rapid development of technology such as high-speed optical communications, wireless mobile communication, and high-performance/low-cost computing, along with the continuous emergence of innovative applications. Cryptography, which deals with different data encryption methodologies, provides a secure way for data communication as the encrypted data are incomprehensible to any third party. Data hiding, also known as steganography, is another way to secure messages during data communication [1], by embedding encrypted secret data in meaningful cover contents, such as a digital image, video, or 3D model, before delivering the stego contents conveying the hidden message to the recipient. Since a data-hiding technique does not change the format of the carrier, it offers additional security to cryptographic techniques. With the combination of both aspects, it provides an additional layer of security for the message during data communication [1].

The ultimate goal of data hiding is to conceal a great number of secret messages within a cover medium and deliver this innocuous stego medium to the recipient in a secured approach without arousing the suspicion of eavesdroppers. Digital images, among a variety of cover contents, are of great interest in the data-hiding community due

to their high popularity and availability; up to three billion images are shared on the Internet daily. Algorithms developed for image data hiding are broadly classified into two categories: irreversible and the reversible methods [1]. The former schemes permit the recipient to extract secret messages conveyed in the stego images. The latter ones, however, enable the recipient not only to extract secret messages but also to restore the stego image, thereby producing the restored image identical to the original cover one.

The least significant bits (LSB) substitution method [2] is a well-known irreversible data-hiding method, which replaces the least significant bits with the hidden secret messages. Matrix embedding [3] and exploiting modification direction (EMD) [4] are two popular irreversible data-hiding algorithms offering high embedding efficiency, and are thereby less vulnerable to steganalytic detection [5,6]. In these irreversible methods, although the original cover images cannot be restored, they actually provide a great amount of embedding capacity, representing the total number of bits that a cover image can carry.

In contrast, reversible data hiding (RDH) enables the receiver to retrieve the original cover image without any image degradation after extracting secret messages, and a number of RDH methods have been proposed in the extant literature. Tian [7] proposed an RDH method based on difference expansion (DE). Ni et al. [8] proposed a histogram shifting- (HS) based data-hiding method. Li et al. [9] proposed a pixel values order (PVO) method. These approaches have boosted the development of RDH algorithms [10–15], aiming to increase the payload or to enhance the image quality.

A conventional RDH scheme usually generates a single stego image. In 2007, Chang et al. [16] proposed an RDH algorithm, where dual stego images were produced to carry two 5-ary secret digits in a cover pixel pair. Reversible data hiding based on such a dual-image approach both increases the payload and distributes the distortion to dual images, therefore attracting attention from researchers [17–29]. Using a larger size of the magic matrix, Chang et al. [17,18] improved the payload reported in [16]. Furthermore, taking advantage of the orientation combinations, Lee and Huang [19] increased the embedding capacity. Qin et al. [20] made use of the EMD method to generate the first stego image and adjusted the modification direction to render the second one. Lu et al. [21] introduced a central folding method, which folded the encoded secret messages into two equal digits before concealing them into two stego images. Lu et al. [22] proposed a similar scheme making use of LSB matching. Yao et al. [23] improved these works [21], offering a larger embedding capacity.

Recently, Lin et al. [26] extended the dual-image RDH scheme to include the turtle shell reference matrix, and Shastri and Thanikaiselvan [27] improved their work by reducing the pixel distortion. Yao et al. [28] utilized the rhombus prediction and error shifting prediction to produce dual stego images with high fidelity. They embedded a half number of the secret bits based on prediction to render a tentative stego image. They then predicted the blank pixels in the tentative stego image to convey the other half number of bits. Although the current state-of-the-art works [30,31] can affectively carry secret messages, providing both satisfactory embedding capacity and stego image quality, they are limited to carrying a single level of secret message.

We consider a new application scenario called "multi-hider reversible data hiding," where several authorized recipients can extract different secret messages using elaborate secret key combinations from dual stego images. In this application, a sender first embeds y different secret data, where each secret message is injected into the cover image by the corresponding secret keys. The produced dual stego images are then uploaded to a cloud stego server or sent to the authorized receivers. The legitimate recipients who hold authorized key(s) can access the cloud stego server and extract one or any parts of the secret messages. The key combination produces L number of message extraction types, where $L = C_1^y + C_2^y + \cdots + C_y^y$. Certainly, the recipient who holds all secret keys denotes the supervisor with the highest rank, able to extract all secret messages. The multi-hider mechanism, together with secret key combinations, provides an advantage to differentially extract assorted secret messages and thus applications therein.

In this paper, we propose a multi-hider dual-image reversible data-hiding scheme (MHDI-RDH) satisfying the above application scenario. Our algorithm can convey three secret messages ($y = 3$) and provide seven levels of secret message extraction ($L = 7$) from dual stego images. The message embedding in our scheme contains two phases. In the first phase, differing from conventional image data-hiding schemes, which utilize the existing image as the cover one to embed messages, our algorithm conceals the first secret message when constructing a new image through an optimal weighted color-transfer process. Specifically, we synthesize a color-transferred image and concurrently convey the first 3-tuple n-ary secret message vector ($\boldsymbol{d_{sb}}$) using the first secret key (key_1), thus producing a tentative stego image. We then embed the second m-ary secret message (d_{wm}) into this tentative stego image and produce a new tentative stego image, I''_{owct}, using the second secret key (key_2). This message-embedding technique is designed elaborately so that the first hidden message survived intact.

Furthermore, in the second phase, we conceal the third 3-tuple k-ary secret message vector ($\boldsymbol{d_{du}}$) with the reversibility benefit using the third secret key (key_3) and produce the final dual stego images, I'''_{s1} and I'''_{s2}. Such a multi-hider approach enables us to independently embed a total of three secret messages, $\boldsymbol{d_{sb}}$, d_{wm}, and $\boldsymbol{d_{du}}$, which, in return, allows the recipients to adopt one of seven types of key combinations to flexibly extract secret messages. This is in contrast to most current state-of-the-art works, which provide the receiver to extract single-level secret data from the dual stego images. For example, an authorized receiver can first extract one secret message vector, $\boldsymbol{d_{sb}}$, from dual stego images and successfully restore the stego image. In addition, if a commissioned secret key is issued, a legitimate receiver with a higher rank of authority can first restore the stego image and then extract the secret messages, d_{wm} and $\boldsymbol{d_{du}}$, from the restored image. To the best of our knowledge, our scheme is the first in the extant literature to offer a multi-hider approach of message embedding, extraction, and stego-image restoration. The experimental results confirm that our scheme enables the associate recipients to gain access to a specific secret message but still ensures that this secret message is only accessible under appropriate security constraints. In addition, our scheme can resist RS and PVD steganalytic attacks. The comparison confirms that the proposed method can achieve high embedding capacity, while maintaining good visual quality, outperforming our counterparts.

We organize this paper as follows. Section 2 introduces the weighted modulus method. In Section 3, we present our proposed method. Section 4 presents our experiment results and analysis. Finally, we offer conclusions in Section 5.

2. An Overview of the Weighted Modulus Algorithm

In this section, we review the weighted modulus algorithm proposed by Chen et al. [32]. Our proposed scheme makes use of the variants of their algorithm to conceal secret messages. Their scheme, denoted by WM(n, m), embeds an m-ary secret message in a pixel cluster with n cover pixels using the modulus operator. Let $\boldsymbol{P}$= (P_1, P_2, … , P_n) be a pixel cluster, s_m be an m-ary secret digit, and $\boldsymbol{W}$= (w_1, w_2, … , w_n) represent an n-tuple weight vector. The stego pixel cluster, $\boldsymbol{P'}$= (p'_1, p'_2, … , p'_n), which has conveyed the hidden digit s_m, is produced through three steps, as detailed below:

Step 1: Referring to Equation (1) to determine the remainder value, r, where "•" represents the dot product of vectors and "mod" represents the modulus operator.
Step 2: Calculating the difference, d, between s_m and r using Equation (2).
Step 3: Inquiring about the pixel alteration vector, $\boldsymbol{A_d}$, from the pixel alteration (PA) table by d before producing $\boldsymbol{P'}$= (p'_1, p'_2, … , p'_n) using Equation (3).

The message extraction is straightforward as we can adopt the dot vector again to obtain s_m, as shown in Equation (4).

$$r = [\boldsymbol{P} \bullet \boldsymbol{W}] \bmod m. \quad (1)$$

$$d = (s_m - r) \mod m. \quad (2)$$

$$\boldsymbol{P}' = \boldsymbol{P} + \boldsymbol{A}_d. \quad (3)$$

$$s_m = [\boldsymbol{P}' \bullet \boldsymbol{W}] \mod m. \quad (4)$$

As an example, let $\boldsymbol{P} = (18, 20, 42)$ be a cover pixel cluster ($n = 3$). Let $s_8 = 2_8$ be an *8*-ary secret digit to be concealed ($m = 8$) and $\boldsymbol{W}$= (7, 5, 2) represent a *3*-tuple weight vector. In step 1, we derive *r*= $[\boldsymbol{P} \bullet \boldsymbol{W}] \mod 8 = 6$. In step 2, we calculate the difference and derive $d = 4$. Finally, we inquire the pixel alteration by $d = 4$ shown in Table 1 and obtain $\boldsymbol{A_4} = (1, 1, 0)$, thus producing the stego pixel cluster $\boldsymbol{P}' = \boldsymbol{P} + \boldsymbol{A}_d = (19, 21, 42)$. The recipient can extract the secret message using $s_8 = [\boldsymbol{P}' \bullet \boldsymbol{W}] \mod 8 = 2_8$.

Table 1. The pixel alteration (PA) table for *3*-tuple weight vector $\boldsymbol{W}$ = (7, 5, 2).

$\boldsymbol{d}$	$\boldsymbol{w_1 = 7}$	$\boldsymbol{w_2 = 5}$	$\boldsymbol{w_3 = 2}$
0	0	0	0
1	−1	0	0
2	0	0	1
3	0	−1	0
4	1	1	0
5	0	1	0
6	0	0	−1
7	1	0	0

The WM algorithm can effectively conceal secret messages using the optimal weight, providing a smaller pixel distortion caused by the message concealment. However, the algorithm does not belong to the approach of reversible data embedding, which means that the original cover image after the data extraction cannot be restored after extracting the secret message. In this paper, we take advantage of the weight modulus and introduce an image reversible data-hiding algorithm, which can provide a high embedding capacity through a multi-hider approach, producing dual stego images, each of which has a high image quality, and retrieving the cover image after the message extraction. We detail our algorithm in the next section.

3. Proposed Method

This section details our proposed algorithm which conceals two levels of secret messages. We first describe the two-phase message embedding on the sender side. The first phase, Phase-1, conceals the first level of secret messages during the process of image synthesis. In Phase-2, we embed the second level of secret message into the synthesized stego image and then produce dual stego images.

On the receiver side, we detail the message extraction, which retrieves the second level of secret message from dual stego images before proceeding to the image restoration and producing the restored image, the synthesized stego image. Finally, the recipient retrieves the first level of secret messages from the restored image. Table 2 lists all the notations used in our algorithm for a quick reference.

Table 2. The notations used in our MHDI-RDH algorithm.

Symbols	Illustration
$\boldsymbol{C_{wm}}$	The distortion vector
$\boldsymbol{V_{sb}}$	The floating-point distortion vector
$\boldsymbol{d_{du}}$	A *k*-ary secret message vector
L	The length of vector

d_{wm}	An m-ary secret message
$\boldsymbol{d_{sb}}$	An n-ary secret message vector
W*	The optimal weight for weighted color transfer
$\boldsymbol{P'''_{s1}}$	The pixel value of the stego image 1
$\boldsymbol{P'''_{s2}}$	The pixel value of the stego image 2
$\boldsymbol{RI''_{owct}}$	The restored image
$\boldsymbol{RP''_{owct}}$	The restored pixel value
$\boldsymbol{FP}$	A single floating-point pixel
I_{sou}	The source image used for optimal weighted color transfer
$\boldsymbol{I''_{owct}}$	The stego OWCT image
$\boldsymbol{P'''_{r1}}$	The stego pair of the red channel
$\boldsymbol{P'''_{g2}}$	The stego pair of the green channel
$\boldsymbol{P'''_{b3}}$	The stego pair of the blue channel
$\boldsymbol{P'_{owct}}$	The tentative stego pixel concealed with one secret message
$\boldsymbol{P''_{owct}}$	The tentative stego pixel concealed with two secret messages
I'''_{s1}	The stego image 1
I'''_{s2}	The stego image 2
I_{tar}	The target image used for optimal weighted color transfer
$\boldsymbol{W_a}$	The 3-tuple weight vector
$\boldsymbol{W_b}$	The 2-tuple weight vector

3.1. Message Embedding

This subsection describes the two-phase message embedding. Figure 1 shows the two-phase message-embedding process, consisting of Phase-1 and Phase-2. The first phase, Phase-1, is denoted as the optimal color-transfer embedding phase, or OWCTE in short. In this phase, a color appearance similarity [33], which utilizes the Kullback–Leibler Divergence (KLD) technique [34], derives an optimal weight, W^*. This optimal weight will be used in OWCT to construct a new image, where its color appearance differs from the source ones. Next, for each pixel, $\boldsymbol{P_{owct}}$, optimal weighted color-transfer embedding (OWCTE) injects the first secret message, an n-ary 3-tuple secret message vector, $\boldsymbol{d_{sb}}$= (d_n^1, d_n^2, d_n^3), into each pixel's Red, Green, and Blue channels. In this phase, we use a secret key, key_1, aiming to achieve a multi-hider mechanism and increase the security of message embedding when producing the stego pixel, $\boldsymbol{P'_{owct}}$.

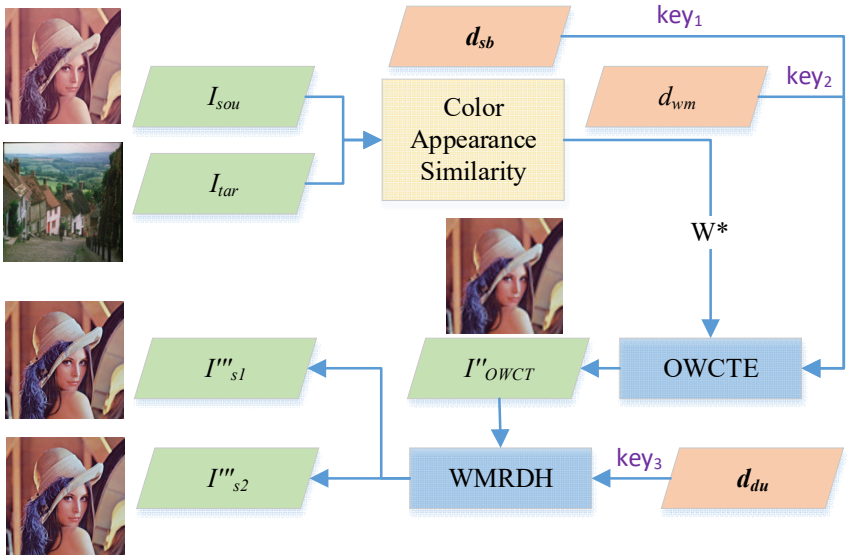


Figure 1. The flowchart of the message embedding in our proposed scheme.

Then, in the second step, a second secret message, d_{wm}, which is in the m-ary notational system, is concealed in every stego pixel, $\boldsymbol{P'_{owct}}$, without affecting the existing hidden message. Similarly, we employ a secret key, key_2, for the purpose of multi-hider and security when producing the stego pixel, $\boldsymbol{P''_{owct}}$. The output of Phase-1 is a stego OWCT image, I''_{owct}, where its color appearance is contributed from both the source and target images with an optimal weight. Every pixel, $\boldsymbol{P''_{owct}}$, has concealed the first secret vector, $\boldsymbol{d_{sb}}$, and the second secret message, d_{wm}.

The second phase, Phase-2, is referred to as the weighed modulus reversible data-hiding phase, or WMRDH in short. In this phase, we embed the third secret message, a k-ary message vector, $\boldsymbol{d_{du}}$ into every stego pixel, $\boldsymbol{P''_{owct}}$. Once again, we use a secret key, key_3, thereby producing dual stego pixels, $\boldsymbol{P'''_{s1}}$ and $\boldsymbol{P'''_{s2}}$. Similarly, we proceed with the message embedding in every pixel, so as to produce dual stego images, I'''_{s1} and I'''_{s2}. The dual stego images enable the receiver to extract one part of the secret data, $\boldsymbol{d_{sb}}$, d_{wm}, and $\boldsymbol{d_{du}}$ independently, and restore the image I''_{owct}, depending on the authorized three secret keys, key_1, key_2, key_3. We detail these two phases as follows.

3.1.1. Phase-1: Optimal Weighted Color-Transfer Embedding (OWCTE)

In this phase, we input the source image, I_{sou}, and the target image, I_{tar}, and conceal two secret messages in two steps when conducting the optimal weighted color transfer. We detail the message embedding in this phase from the viewpoint of a single pixel. Algorithm 1 shows the OWCTE, where $\boldsymbol{FP}$= (r, g, b) denotes a single floating-point pixel, represented by three floating-point values, where all of them are within the range of [0.0, 255.0]. Readers are recommended to refer to [33] for a detailed description about the optimal color-transfer process. Here, we describe the process of producing a single floating-point pixel in brief, and adopt the Red channel as an example. We assume that an optimal weight, W^*, has been determined. Given a pixel, $P_R(x, y)$, in the source image at the position (x, y), we can produce a color-transferred pixel with the floating-point value, $FP_R(x, y)$, in the Red channel by Equation (5), where σ_s and σ_t indicate the respective standard deviations of the source and target image in the Red channel, and μ_s and μ_t denote the means of the source and target image in the Red channel, respectively.

$$FP_R(x,y) = \frac{W^*\sigma_t + (1-W^*)\sigma_s}{\sigma_s}(P_R(x,y) - \mu_s) + W^*\mu_t + (1-W^*)\mu_s \quad (5)$$

As an example, we consider a pixel in the position $(x, y) = (18, 33)$, where the pixel in the Red channel has the value $P_R(18, 33) = 14$. Let $W^* = 0.493$, $\sigma_s = 24.56$, $\sigma_t = 23.89$, $\mu_s = 120.7$, and $\mu_t = 124.5$. Given these parameters, we can adopt Equation (5) to derive $FP_R(18, 33) = 17.3$. We can calculate the floating-point pixel values in the Green and Blue channels accordingly, thereby producing the available $\boldsymbol{FP}$= (r, g, b) for Algorithm 1. Note that $\boldsymbol{FP}$= (r, g, b) is produced in the mid-process of the optimal color transfer when pixels have not been rounded up or down to produce the integer values.

Algorithm 1. OWCTE algorithm

Input: $\boldsymbol{FP}$= (r, g, b), $\boldsymbol{W_a}$ = [w_1, w_2, w_3], $\boldsymbol{d_{sb}}$= (d_n^1, d_n^2, d_n^3), and d_{wm}

Output: $\boldsymbol{P''_{owct}}$

1. Determine the minimal distortion vector, $\boldsymbol{V_{sb}}$ = (v_r, v_g, v_b), to satisfy Equation (6).
2. Determine the coefficient, $\boldsymbol{C_{wm}}$= (c_r, c_g, c_b), to satisfy Equations (7) and (8).
3. Produce $\boldsymbol{P''_{owct}}$ = $\boldsymbol{FP}$+$\boldsymbol{V_{sb}}$+n$\boldsymbol{C_{wm}}$.

Let $\boldsymbol{P'_{owct}}$= (r', g', b') be the stego pixel which has a concealed n-ary secret message vector, $\boldsymbol{d_{sb}}$. Let $\boldsymbol{P''_{owct}}$ represent the stego pixel produced in Phase-1, which has both $\boldsymbol{d_{sb}}$ and d_{wm} concealed. The OWCTE algorithm describes the message embedding in this phase, which consists of three steps.

Step 1: Producing the stego pixel $\boldsymbol{P'_{owct}} = (r',\ g',\ b')$. Given $\boldsymbol{FP}$= (r, g, b), we determine the floating-point distortion vector, $\boldsymbol{V_{sb}} = (v_r,\ v_g,\ v_b)$, such that $\boldsymbol{P'_{owct}}$ mod $n = (d_n^1,\ d_n^2,\ d_n^3)$ and $\boldsymbol{V_{sb}}$ have the minimal length, $L_{min} = [(v_r)^2+(v_g)^2+(v_b)^2]^{0.5}$, as shown in Equation (6), where $\boldsymbol{P'_{owct}} = \boldsymbol{FP}+\boldsymbol{V_{sb}}$. Note that there are a number of distortion vectors which satisfy the above modulus operation. We determine the one that has the minimal length in aiming to reduce the pixel modification to as small as possible.

Example-1: If $\boldsymbol{FP}$ = (17.3, 3.7, 20.6) and $\boldsymbol{d_{sb}} = (1_3,\ 2_3,\ 0_3)$ represents a 3-ary message vector (n = 3) to be concealed. We determine $\boldsymbol{V_{sb}}$ = (−1.3, 1.3, 0.4), leading to $\boldsymbol{P'_{owct}} = \boldsymbol{FP}+\boldsymbol{V_{sb}} = (r',\ g',\ b')$= (16, 5, 21). Clearly, $\boldsymbol{P'_{owct}}$ mod n = (16 mod 3, 5 mod 3, 21 mod 3) = (1, 2, 0) satisfies Equation (6). In addition, $\boldsymbol{V_{sb}}$ has the minimal length, where L_{min} = $[(-1.3)^2 + (1.3)^2 + (0.4)^2]^{0.5}$. If we instead determine $\boldsymbol{V_{sb}}$ = (1.7, 1.3, 0.4), this leads to $\boldsymbol{P'_{owct}}$ = (19, 5, 21). Although $\boldsymbol{P'_{owct}}$ satisfies the modulus operation, the length of $\boldsymbol{V_{sb}}$, denoted as L, is not minimal because $L = [(1.7)^2 + (1.3)^2 + (0.4)^2]^{0.5} > L_{min}$.

Step 2: Producing the final stego pixel $\boldsymbol{P''_{owct}} = (r'',\ g'',\ b'')$. We determine the distortion vector $\boldsymbol{C_{wm}} = (c_r,\ c_g,\ c_b)$ such that $\boldsymbol{P''_{owct}}$ satisfies the following two conditions. First, it complies with the modulus operation, ($\boldsymbol{P''_{owct}}$ dot $\boldsymbol{W_a}$) mod m= d_{wm} and $n \times \boldsymbol{C_{wm}}$ has minimal length, $L_{min} = [\, n^2\,(c_r^2 + c_g^2 + c_b^2\,)\,]^{0.5}$ as shown in Equation (7), where $\boldsymbol{P''_{owct}} = \boldsymbol{P'_{owct}} + n \times \boldsymbol{C_{wm}}$. Note $\boldsymbol{W_a} = [w_1^a,\ w_2^a,\ w_3^a]$ represents the weight vector given by the end user for modulus operation. Any distortion vectors satisfying Equation (7) indicate that the stego pixel $\boldsymbol{P''_{owct}}$ has a concealed secret message, d_{wm}, which can later be extracted on the receiver side. The second condition, shown in Equation (8), states that $\boldsymbol{P''_{owct}}$ must be within the eligible range, [LB(k), UB(k)], where LB(k) and UB(k) represent the lower bound and upper bound of the range, respectively, The eligible range depends on parameter k, which represents the notational system based on which a secret message will be concealed in Phase-2. This condition ensures that the current stego pixel, $\boldsymbol{P''_{owct}}$, is eligible to conceal a k-ary secret message in Phase-2 without affecting previous hidden messages and encountering an overflow or underflow problem.

Specifically, for example, if we intend to conceal a 3-ary secret message in Phase-2 (k = 3), the pixel range is restricted to [0, 254], because embedding such a secret message will cause a pixel change in the range of [0, +1]; thereby, the lower bound is 0 and the upper bound is 254. If we instead convey a 5-ary secret message (k = 5), the pixel change is in the range [1, 254] because the maximal magnitude of change due to message concealment is plus or minus 1 ($\pm$1).

Following up on Example-1, a stego pixel, $\boldsymbol{P'_{owct}}$ = (16, 5, 21), has been produced after concealing the 3-ary message vector, $\boldsymbol{d_{sb}} = (1_3,\ 2_3,\ 0_3)$. Let $d_{wm} = 2_7$ represent a 7-ary secret message to be concealed in this step (m = 7). Let $\boldsymbol{W_a}$= [1, 2, 3] represent the weight vector for message concealment. Without loss of generality, we assume the secret message to be conveyed in Phase-2 is in the 5-ary notational system (k = 5); thereby, the eligible range is [1, 254]. To conceal the secret message, $d_{wm} = 2_7$, we determine $\boldsymbol{C_{wm}}$ = (−1, 0, 0) producing $\boldsymbol{P''_{owct}}$ = (16, 5, 21) + 3*(−1, 0, 0) = (13, 5, 21). Clearly, $\boldsymbol{P''_{owct}}$ satisfies Equation (7) because (13, 5, 21) dot [1, 2, 3] mod 7 = 2_7 and $\boldsymbol{C_{wm}}$ has the minimal length, L_{min}= 1.

$$\boldsymbol{P'_{owct}} \text{ mod } n = (d_n^1,\ d_n^2,\ d_n^3), \text{ where } \boldsymbol{P'_{owct}} = \boldsymbol{FP}+\boldsymbol{V_{sb}} \tag{6}$$

$$(\boldsymbol{P''_{owct}} \text{ dot } \boldsymbol{W_a}) \text{ mod } m = d_{wm}, \text{ where } \boldsymbol{P''_{owct}} = \boldsymbol{P'_{owct}} + n \times \boldsymbol{C_{wm}} \tag{7}$$

$$LB(k) \le \boldsymbol{P''_{owct}} \le UB(k) \tag{8}$$

3.1.2. Phase-2: Weight Modulus Reversible Data-Hiding Embedding (WMRDHE)

In Phase-2, we input a stego OWCT image, $\boldsymbol{I''_{owct}}$, where every stego pixel, $\boldsymbol{P''_{owct}} = (r'',\ g'',\ b'')$, has one secret message vector concealed, $\boldsymbol{d_{sb}}$, and one secret message conveyed, d_{wm}.

Algorithm 2 shows the weight modulus reversible data-hiding embedding. The output of Phase-2 is two stego images: I_{s1}''' and I_{s2}''', which are produced by concealing a number of k-ary secret message vectors, $\boldsymbol{d_{du}} = (d_k^1,\ d_k^2,\ d_k^3)$, using the modulus operation with the 2-tuple weight vector, $\boldsymbol{W_b} = [w_1,\ w_2]$. The message embedding consists of three steps, as detailed below:

Algorithm 2. WMRDH algorithm

Input: $\boldsymbol{I_{owct}''}$, $\boldsymbol{W_b} = [w_1,\ w_2]$, and $\boldsymbol{d_{du}} = (d_k^1,\ d_k^2,\ d_k^3)$

Output: I_{s1}''' and I_{s2}'''

1. Constructing the "dual-pixel," $\boldsymbol{P_{du}''}$.
2. Locating the distortion vector, $\boldsymbol{V_{d_u}}$, from Equation (9).
3. Producing the stego pair of components in three channels, $\boldsymbol{P_{r1}'''}$, $\boldsymbol{P_{g2}'''}$, and $\boldsymbol{P_{b3}'''}$, from Equation (10), and producing the dual stego images: I_{s1}''' and I_{s2}'''.

Step 1: Constructing the "dual-pixel." In this step, we first duplicate two images from $\boldsymbol{I_{owct}''}$ and rename them as I_{s1}'' and I_{s2}'', respectively. Without loss of generality, we denote a pixel in I_{s1}'' and I_{s2}'' as $\boldsymbol{P_{s1}''} = (r_{s1}'',\ g_{s1}'',\ b_{s1}'')$ and $\boldsymbol{P_{s2}''} = (r_{s2}'',\ g_{s2}'',\ b_{s2}'')$, respectively. Then, we construct a dual-pixel, $\boldsymbol{P_{du}''} = ((r_{s1}'',\ r_{s2}''),\ (g_{s1}'',\ g_{s2}''),\ (b_{s1}'',\ b_{s2}''))$, from $\boldsymbol{P_{s1}''}$ and $\boldsymbol{P_{s2}''}$. In other words, we select the component in the Red channel from $\boldsymbol{P_{s1}''}$ in I_{s1} and $\boldsymbol{P_{s2}''}$ in I_{s2}, to construct a Red component pair, $(r_{s1}'',\ r_{s2}'')$. Similarly, we proceed with the same construction approach for the components in the Green and Blue channels. To simplify the notation, we represent each component in $\boldsymbol{P_{du}''}$ as $\boldsymbol{P_{r1}''} = (r_{s1}'',\ r_{s2}'')$, $\boldsymbol{P_{g2}''} = (g_{s1}'',\ g_{s2}'')$, $\boldsymbol{P_{b3}''} = (b_{s1}'',\ b_{s2}'')$. For example, if $\boldsymbol{P_{owct}''} = (r'',\ g'',\ b'') = (13, 5, 21)$, the dual-pixel constructed is $\boldsymbol{P_{du}''} = ((13, 13), (5, 5), (21, 21))$, and the component pairs in the Red, Green, and Blue channels will be $\boldsymbol{P_{r1}''} = (13, 13)$, $\boldsymbol{P_{g2}''} = (5, 5)$, and $\boldsymbol{P_{b3}''} = (21, 21)$, respectively.

Step 2: Locating the distortion vector, $\boldsymbol{V_{d_k}}$. We use an individual secret message in $\boldsymbol{d_{du}}$ as the row index and the weight vector, $\boldsymbol{W_b} = [w_1,\ w_2]$, as the color index to derive the corresponding 2-tuple distortion vector, $\boldsymbol{V_{d_k}}$, stored in the distortion vector table (DVT). For example, let $\boldsymbol{d_{du}} = (d_k^1,\ d_k^2,\ d_k^3) = (3_5,\ 2_5,\ 4_5)$ be the 5-ary secret message vector (k = 5) and $\boldsymbol{W_b} = [1,4]$ be the weight vector. The distortion vector table, shown in Table 3, indicates that $\boldsymbol{V_3} = [-1, 1]$ is the distortion vector corresponding to 3_5. Similarly, the distortion vectors corresponding to 2_5 and 4_5 are $\boldsymbol{V_2} = [1, -1]$ and $\boldsymbol{V_4} = [0, 1]$, respectively. We simplify the process in Step 2 as the DVT function, where its entry is a parameter pair, $(d_k,\ \boldsymbol{W_b})$, and the output is the distortion vector, $\boldsymbol{V_{d_k}}$, as show in Equation (9).

Table 3. The distortion vector table (DVT) for concealing 5-ary secret messages.

	$\boldsymbol{W_b}$ **= [1, 4]**	$\boldsymbol{W_b}$ **= [2, 3]**
d_5	V_{d_5}	V_{d_5}
0	$\boldsymbol{V_0} = [0, 0]$	$\boldsymbol{V_0} = [0, 0]$
1	$\boldsymbol{V_1} = [1, 0]$	$\boldsymbol{V_1} = [-1, 1]$
2	$\boldsymbol{V_2} = [1, -1]$	$\boldsymbol{V_2} = [1, 0]$
3	$\boldsymbol{V_3} = [-1, 1]$	$\boldsymbol{V_3} = [0, 1]$
4	$\boldsymbol{V_4} = [0, 1]$	$\boldsymbol{V_4} = [1, -1]$

Step 3: Producing the stego component pairs in three channels, $\boldsymbol{P_{r1}'''}$, $\boldsymbol{P_{g2}'''}$, and $\boldsymbol{P_{b3}'''}$. Using the vector addition, shown in Equation (10), we can embed a secret message vector $\boldsymbol{d_{du}}$ into the dual-pixel $\boldsymbol{P_{du}''}$ and produce the stego dual-pixel, $\boldsymbol{P_{du}'''}$, which contains the stego component pairs, $\boldsymbol{P_{r1}'''}$, $\boldsymbol{P'''_{g2}}$, and $\boldsymbol{P'''_{b3}}$.

Following up on Example-1, we have constructed the dual-pixel, $\boldsymbol{P_{du}''} = ((13, 13), (5, 5), (21, 21))$. Let $\boldsymbol{d_{du}} = (3_5,\ 2_5,\ 4_5)$ be the secret message vector, and $\boldsymbol{V_3} = [-1, 1]$, $\boldsymbol{V_2} = [1, -1]$, and $\boldsymbol{V_4} = [0, 1]$ represent the corresponding distortion vectors. We can produce the stego component pair, $\boldsymbol{P_{r1}'''} = (13, 13) + (-1, 1) = (12, 14)$. Similarly, we can produce the other

two stego component pairs, $\boldsymbol{P}'''_{\boldsymbol{g2}}$ = (5, 5) + (1, −1)= (6, 4) and $\boldsymbol{P}'''_{\boldsymbol{b3}}$= (21, 21) + (0, 1) = (21, 22). Therefore, the stego dual-pixel produced becomes $\boldsymbol{P}'''_{\boldsymbol{du}}$= ((12, 14), (6, 4), (21, 22)). Note that the corresponding stego pixels in I'''_{s1} and I'''_{s2} are $\boldsymbol{P}'''_{\boldsymbol{s1}}$= (12, 6, 21) and $\boldsymbol{P}'''_{\boldsymbol{s2}}$ = (14, 4, 22), respectively. We remark that in Phase-1, these two stego pixels have a 3-ary secret message vector, $\boldsymbol{d}_{\boldsymbol{sb}}$ = (1_3, 2_3, 0_3), concealed, and a 7-ary secret message, 2_7. In addition, in Phase-2, they have also conveyed a 5-ary secrete message vector, $\boldsymbol{d}_{\boldsymbol{du}}$= ($3_5$, 2_5, 4_5). Clearly, we can produce the stego component pairs for every pixel in three channels: $\boldsymbol{P}'''_{\boldsymbol{r1}}$, $\boldsymbol{P}'''_{\boldsymbol{g2}}$, and $\boldsymbol{P}'''_{\boldsymbol{b3}}$, by a vector addition shown in Equation (10). Consequently, dual stego images, I'''_{s1} and I'''_{s2}, will be produced in Phase-2.

$$\boldsymbol{V}_{\boldsymbol{d_k}} = \mathrm{DVT}(d_k, \boldsymbol{W_b}) \tag{9}$$

$$\begin{cases} \boldsymbol{P}'''_{\boldsymbol{r1}} = \boldsymbol{P}''_{\boldsymbol{r1}} + \mathrm{DVT}(d_k^1, \boldsymbol{W_b}) \\ \boldsymbol{P}'''_{\boldsymbol{g2}} = \boldsymbol{P}''_{\boldsymbol{g2}} + \mathrm{DVT}(d_k^2, \boldsymbol{W_b}) \\ \boldsymbol{P}'''_{\boldsymbol{b3}} = \boldsymbol{P}''_{\boldsymbol{b3}} + \mathrm{DVT}(d_k^3, \boldsymbol{W_b}) \end{cases} \tag{10}$$

Tables 4 and 5 show the distortion vector tables for concealing the 7-ary and 9-ary secret messages, respectively. Each table contains three weight vectors ($\boldsymbol{W_b}$), which serve as an entry for the function, DVT(), to enquire the distortion vector, $\boldsymbol{V}_{\boldsymbol{d_k}}$, shown in Equation (9).

Table 4. The distortion vector table for concealing 7-ary secret messages.

	$\boldsymbol{W_b}$ **= [1, 6]**	$\boldsymbol{W_b}$ **= [2, 5]**	$\boldsymbol{W_b}$ **= [3, 4]**
d_7	V_{d_7}	V_{d_7}	V_{d_7}
0	$\boldsymbol{V_0}$ = [0, 0]	$\boldsymbol{V_0}$ = [0, 0]	$\boldsymbol{V_0}$ = [0, 0]
1	$\boldsymbol{V_1}$ = [1, 0]	$\boldsymbol{V_1}$ = [−1, 2]	$\boldsymbol{V_1}$ = [−1, 1]
2	$\boldsymbol{V_2}$ = [1, −1]	$\boldsymbol{V_2}$ = [1, 0]	$\boldsymbol{V_2}$ = [2, −1]
3	$\boldsymbol{V_3}$ = [2, −1]	$\boldsymbol{V_3}$ = [−1, 1]	$\boldsymbol{V_3}$ = [1, 0]
4	$\boldsymbol{V_4}$ = [−1, 2]	$\boldsymbol{V_4}$ = [1, −1]	$\boldsymbol{V_4}$ = [0, 1]
5	$\boldsymbol{V_5}$ = [−1, 1]	$\boldsymbol{V_5}$ = [0, 1]	$\boldsymbol{V_5}$ = [−1, 2]
6	$\boldsymbol{V_6}$ = [0, 1]	$\boldsymbol{V_6}$ = [2, −1]	$\boldsymbol{V_6}$ = [1, −1]

Table 5. The distortion vector table for concealing 9-ary secret messages.

	$\boldsymbol{W_b}$ **= [1, 8]**	$\boldsymbol{W_b}$ **= [2, 7]**	$\boldsymbol{W_b}$ **= [4,5]**
d_9	V_{d_9}	V_{d_9}	V_{d_9}
0	$\boldsymbol{V_0}$ = [0, 0]	$\boldsymbol{V_0}$= [0, 0]	$\boldsymbol{V_0}$= [0, 0]
1	$\boldsymbol{V_1}$ = [1, 0]	$\boldsymbol{V_1}$ = [−2, 2]	$\boldsymbol{V_1}$= [−1, 1]
2	$\boldsymbol{V_2}$ = [1, −1]	$\boldsymbol{V_2}$ = [1, 0]	$\boldsymbol{V_2}$= [−2, 2]
3	$\boldsymbol{V_3}$ = [2, −1]	$\boldsymbol{V_3}$ = [−1, 2]	$\boldsymbol{V_3}$= [2, −1]
4	$\boldsymbol{V_4}$ = [2, −2]	$\boldsymbol{V_4}$ = [1, −1]	$\boldsymbol{V_4}$= [1, 0]
5	$\boldsymbol{V_5}$ = [−2, 2]	$\boldsymbol{V_5}$ = [−1, 1]	$\boldsymbol{V_5}$= [0, 1]
6	$\boldsymbol{V_6}$ = [−1, 2]	$\boldsymbol{V_6}$ = [2, −1]	$\boldsymbol{V_6}$= [−1, 2]
7	$\boldsymbol{V_7}$ = [−1, 1]	$\boldsymbol{V_7}$ = [0, 1]	$\boldsymbol{V_7}$= [2, −2]
8	$\boldsymbol{V_8}$ = [0, 1]	$\boldsymbol{V_8}$ = [2, −2]	$\boldsymbol{V_8}$= [1, −1]

3.1.3. An Example of Message Embedding

In this subsection, we describe Example-2, an example of message embedding. In Phase-1, let $\boldsymbol{FP}$ = (95.9, 138.0, 155.1) denote a pixel represented by three floating-point values during the mid-phase of optimal color transfer. Let $\boldsymbol{d}_{\boldsymbol{sb}}$= ($1_3$, 0_3, 2_3) denote the 3-ary secret message vector to be concealed. First, we determine the minimum distortion vector, $\boldsymbol{V}_{\boldsymbol{sb}}$ = (1.1, 0.0, −0.1), where $\boldsymbol{P}'_{\boldsymbol{owct}}$ = $\boldsymbol{FP}$+$\boldsymbol{V}_{\boldsymbol{sb}}$= (97, 138, 155) satisfies Equation (6), implying that $\boldsymbol{P}'_{\boldsymbol{owct}}$ has embedded a secret message vector, $\boldsymbol{d}_{\boldsymbol{sb}}$. Next, let d_{wm}= 2_7

represent the 7-ary secret message to be concealed, and $\boldsymbol{W_a}$= (1, 2, 3) be the weight vector. We determine the coefficient $\boldsymbol{C_{wm}}$ = (0, 0, 1), which has the minimal length L_{min} = 1 and $\boldsymbol{P'_{owct}}$+3 × $\boldsymbol{C_{wm}}$ satisfies both Equations (7) and (8). In this phase, we produce $\boldsymbol{P''_{owct}}$= (97, 138, 158), which has conveyed $\boldsymbol{d_{sb}}$ and d_{wm}.

In Phase-2, let $\boldsymbol{d_{du}}$ = (1_5, 3_5, 0_5) be the 5-ary secret message vector to be conveyed, and $\boldsymbol{W_b}$= (1, 4) be the weight vector. From Equation (9) and Table 3 , we derive r distortion vectors $\boldsymbol{V_1}$= (1, 0), $\boldsymbol{V_3}$= (−1, 1), and $\boldsymbol{V_0}$ = (0, 0) for the secret messages, 1_5, 3_5, and 0_5, respectively. Finally, we apply Equation (10) and produce three stego component pairs: $\boldsymbol{P'''_{r1}}$= [97, 97] + [1, 0] = [98, 97], $\boldsymbol{P'''_{g2}}$= [138, 138] + [−1, 1] = [137, 139], and $\boldsymbol{P'''_{b3}}$ = [158, 158] + [0, 0] = [158, 158]. Consequently, the stego pixels in dual stego images, I'''_{s1} and I'''_{s2}, are $\boldsymbol{P'''_{s1}}$= (98, 138, 158) and $\boldsymbol{P'''_{s2}}$ = (97, 139, 158), respectively.

3.2. Message Extraction Algorithm

This section describes the message extraction, consisting of two phases, as shown in Figure 2. The message extraction is in a reverse order. In Phase-1, the recipient inputs dual stego images, I'''_{s1} and I'''_{s2}, the weight vector, $\boldsymbol{W_b}$ = [w_1, w_2], and the authorized secret key, key_3, to extract the latest *k*-ary secret message vector, $\boldsymbol{d_{du}}$= (d_k^1, d_k^2, d_k^3), from every dual-pixel. Note that it is not necessary to restore the secret image in order to extract the message vector, $\boldsymbol{d_{du}}$.

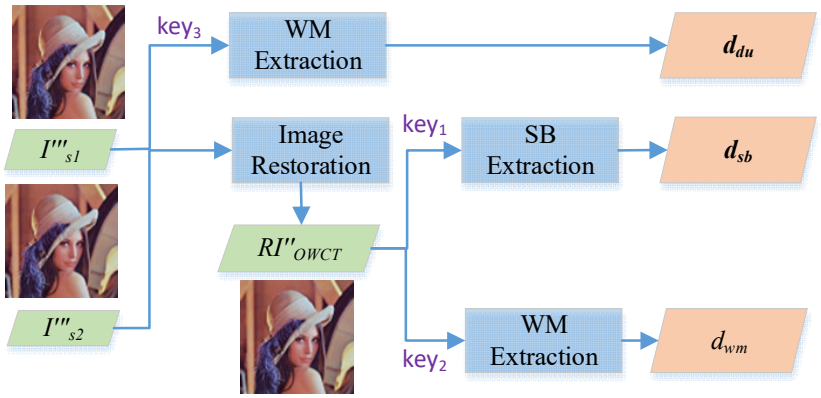


Figure 2. The flowchart of message extraction.

Next, in Phase-2, the pixels in dual stego images are employed to restore the stego pixel, $\boldsymbol{RP''_{owct}}$, using a simple, yet efficient addition and floor function, as shown in Equation (12). Next, the recipient who holds the secret key, key_2, can independently extract the *m*-ary secret message, d_{wm}, using a 3-tuple weight vector, $\boldsymbol{W_a}$ = [w_1, w_2, w_3]. Finally, the *n*-ary secret message vector, $\boldsymbol{d_{sb}}$ = (d_n^1, d_n^2, d_n^3), is extracted by the legitimate recipient who holds the secret key, key_1. We now detail each phase as the following.

3.2.1. Message Extraction by Weighted Modulus and Image Restoration

In this phase, dual stego images are processed to extract the message vector, $\boldsymbol{d_{du}}$ = (d_k^1, d_k^2, d_k^3) concealed in pixels, $\boldsymbol{P'''_{s1}}$= (r'''_{s1}, g'''_{s1}, b'''_{s1}) in I'''_{s1} and $\boldsymbol{P'''_{s2}}$= (r'''_{s2}, g'''_{s2}, b'''_{s2}) in I'''_{s2}, respectively. First, we construct a dual-pixel, $\boldsymbol{P'''_{du}}$= ((r'''_{s1}, r'''_{s2}), (g'''_{s1}, g'''_{s2}), (b'''_{s1}, b'''_{s2})) from $\boldsymbol{P'''_{s1}}$ and $\boldsymbol{P'''_{s2}}$ and represent each component pair as $\boldsymbol{P'''_{r1}}$= (r'''_{s1}, r'''_{s2}), $\boldsymbol{P'''_{g2}}$= (g'''_{s1}, g'''_{s2}) $\boldsymbol{P'''_{b3}}$= (b'''_{s1}, b'''_{s2}). We then input the corresponding 2-tuple weight vectors, $\boldsymbol{W_b}$= [w_1, w_2] and the parameter *k* for each component pair to extract the secret message vector, $\boldsymbol{d_{du}}$, by Equation (11). Finally, we restore the pixel, thereby producing the recovered pixel, $\boldsymbol{RP''_{owct}}$, from each component pair in the dual-pixel, $\boldsymbol{P'''_{du}}$, using Equation (12), where ⌊ ⌋ represents the floor function. We can certainly produce the recovered optimal color-transferred image, RI''_{owct}, after restoring every pixel. We remark that since the recovered pixel, $\boldsymbol{RP''_{owct}}$, is identical to the stego pixel, $\boldsymbol{P''_{owct}}$, produced in the sender part, the recovered

image, RI''_{owct}, is also indistinguishable from the stego image, I''_{owct}, produced in the sender part, thereby implying that our scheme can fully restore the stego image.

$$\boldsymbol{d_{du}} = (d_k^1,\ d_k^2,\ d_k^3) = ([\boldsymbol{P'''_{r1}} \cdot \boldsymbol{W_b}] \bmod k,\ [\boldsymbol{P'''_{g2}} \cdot \boldsymbol{W_b}] \bmod k,\ [\boldsymbol{P'''_{b3}} \cdot \boldsymbol{W_b}] \bmod k) \quad (11)$$

$$\boldsymbol{RP''_{owct}} = (\left\lfloor \frac{r'''_{s1} + r'''_{s2}}{2} \right\rfloor,\ \left\lfloor \frac{g'''_{s1} + g'''_{s2}}{2} \right\rfloor,\ \left\lfloor \frac{b'''_{s1} + b'''_{s2}}{2} \right\rfloor) \quad (12)$$

3.2.2. Message Extraction by Single Base and Weight Modulus

In this phase, the recovered image, RI''_{owct}, is processed pixel-by-pixel to extract messages hidden within it. Given RI''_{owct}, we process each pixel, $\boldsymbol{RP''_{owct}}$, and derive the m-ary secret message, d_{wm}, using the 3-tuple weight vector, $\boldsymbol{W_a}$, and modulus operation, as shown in Equation (13). We then extract the secret message vector, $\boldsymbol{d_{sb}}$, using the parameter n and the modulus operation, as shown in Equation (14).

$$d_{wm} = \boldsymbol{RP''_{owct}} \cdot \boldsymbol{W_a} \bmod m \quad (13)$$

$$\boldsymbol{d_{sb}} = (d_{sb}^1,\ d_{sb}^2,\ d_{sb}^3) = \boldsymbol{RP''_{owct}} \bmod n \quad (14)$$

We next describe an example of message extraction. Following up on Example-2, the recipient inputs the pixels $\boldsymbol{P'''_{s1}}$ = (98, 138, 158) and $\boldsymbol{P'''_{s2}}$ = (97, 139, 158) in dual stego images, I'''_{s1} and I'''_{s2}, respectively. The recipient acquires the 3-tuple weight vector, $\boldsymbol{W_a}$= [1, 2, 3] and the 2-tpple weight vector, $\boldsymbol{W_b}$ = [1, 4] as secret keys delivered in prior by the sender, along with parameters n = 3, m = 7, k = 5. The secret messages are extracted in reverse order: $\boldsymbol{d_{du}}$, first, followed by d_{wm}, and finally $\boldsymbol{d_{sb}}$. First, the recipient extracts the secret message vector, $\boldsymbol{d_{du}}$, using Equation (11), thus producing $\boldsymbol{d_{du}}$= (1_5, 3_5, 0_5). Next, an image is restored, and using Equation (12), the recovered pixel is produced from $\boldsymbol{P'''_{s1}}$ and $\boldsymbol{P'''_{s2}}$, leading to $\boldsymbol{RP''_{owct}}$= (97, 138, 158). Furthermore, the recipient extracts the 7-ary secret message using Equation (13) and the 3-tuple weight vector $\boldsymbol{W_a}$, thus producing d_{wm}= (13, 5, 21)·(1, 2, 3) mod 7 = 2_7. Finally, the recipient extracts the 3-ary secret message vector using Equation (14), thus producing $\boldsymbol{d_{sb}}$= (1_3, 0_3, 2_3). All of the other secret messages can be extracted in a similar approach from the corresponding pixels in dual stego images, I'''_{s1} and I'''_{s2}, respectively.

3.3. Complexity Analysis and Hierarchical Levels for Message Extraction

We now analyze the time complexity of our scheme. Let $W \times H$ represents the size of the image. In the message embedding, we adopt the OWCTE algorithm in Phase-1 and the WMRDH algorithm in Phase-2. Both algorithms embed the respective secret vector and message, thus requiring the time complexity of $O(W \times H)$. Similarly, in the message-extraction process, we adopt similar algorithms but in a reverse order. Therefore, the time complexity is $O(W \times H)$. In summary, the time complexity of our proposed scheme is $O(W \times H)$. Our proposed algorithm provides a multi-hider reversible data-hiding mechanism. In particular, the sender sends dual stego images concealing three secret messages to recipients, in a hierarchical way. On the receiving side, a receiver who holds authorized secret key(s) can extract one or several parts of the secret data independently.

We analyze the secret keys required by the authorized recipient to extract secret messages in Table 6. A total of seven levels for message extraction are available for a combination of three secret keys, key_3, key_2, and key_1. As an example, a receiver has the lowest rank if a secret key is assigned; for example, the secret message vector, $\boldsymbol{d_{du}}$, can be extracted by a receiver in level 1 who has a single secret key, key_3, without even restoring the stego image. In contrast, a receiver holding three secret keys, key_3, key_2, and key_1, has the highest rank (level 7), able to extract the entire secret messages. Our proposed algorithm offers two benefits due to the multi-hider mechanism. First, it enables application access to be varied under varying and dynamic secret key operating conditions,

preferably in a sender-specific customized manner. Second, the secret keys and their hierarchical forms can be applied and then modified appropriately based on changing circumstances, for example, in the cloud server, thereby providing additional security guarantees.

Table 6. The seven levels of message extraction for recipients holding authorized secret keys.

Level	Required Secret Keys	Extracted Messages	Image Restoration
1	key_3	$\boldsymbol{d_{du}}$	No Need
2	key_2	d_{wm}	RI''_{owct}
3	key_1	$\boldsymbol{d_{sb}}$	RI''_{owct}
4	$key_3,\ key_2$	$\boldsymbol{d_{du}},\ d_{wm}$	RI''_{owct}
5	$key_3,\ key_1$	$\boldsymbol{d_{du}},\ \boldsymbol{d_{sb}}$	RI''_{owct}
6	$key_2,\ key_1$	$d_{wm},\ \boldsymbol{d_{sb}}$	RI''_{owct}
7	$key_3,\ key_2,\ key_1$	$\boldsymbol{d_{du}},\ d_{wm},\ \boldsymbol{d_{sb}}$	RI''_{owct}

4. Experiment and Security Analysis

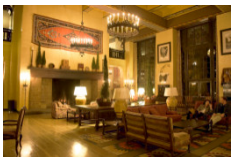
We present the experiment results in this section. We implemented our algorithm using C++ and Python programming languages and collected the experiment results in a personal computer equipped with 2.40 GHz CPU and 8 GB memory. Our test datasets include the color version of the tone-mapped color image from the HDR image database (Figures 3 and 4), USC-SIPI image database (Figures 5–7), and the Kodak image database (Figures 8–10). In this section, we first present the visual inspection for images generated by our algorithm. We then analyze the embedding capacity followed by reporting the image quality statistics. Next, we show the RS steganalytic results and the pixel value difference (PVD) result to demonstrate that our algorithm can resist malicious attacks from eavesdroppers. Furthermore, we present the robustness analysis when a stego image is under salt-and-pepper or cropping attacks. Finally, we compare our scheme with current state-of-the-art works to demonstrate that our scheme is superior to others.

Source, I_{sou}

I_{tar}: Cemetery Tree (1) I''_{owct}, W* = (0.58, 0.60, 0.71) dual stego image, I'''_{s1} dual stego image, I'''_{s2}

Figure 3. The visual inspection of two cases of stego images, "General Sherman," produced by our scheme.

Source, I_{sou}



I_{tar}: Ahwahnee Great Lounge

I''_{owct}, W* = (0.51, 0.47, 0.52)

dual stego image, I'''_{s1}

dual stego image, I'''_{s2}

Figure 4. The visual inspection of the stego images, "Clock Building," produced by our scheme.

I_{sou}

I_{tar}

I''_{owct}, W* = (0.45, 0.48, 0.50)

dual stego image, I'''_{s1}

dual stego image, I'''_{s2}

Figure 5. The visual inspection of the test image "Baboon".

I_{sou}

I_{tar}

I''_{owct}, W* = (0.48, 0.60, 0.42)

dual stego image, I'''_{s1}

dual stego image, I'''_{s2}

Figure 6. The visual inspection of the test image "Lena".

I_{sou}

I_{tar}

I''_{owct}, W* = (0.47, 0.54, 0.48) dual stego image, I'''_{s1} dual stego image, I'''_{s2}

Figure 7. The visual inspection of the test image "Pepper".

I_{sou}

I_{tar}

I''_{owct}, W* = (0.48, 0.46, 0.48) dual stego image, I'''_{s1} dual stego image, I'''_{s2}

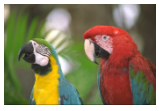
Figure 8. The visual inspection of the test image "Kodim07".

I_{sou}

I_{tar}

I''_{owct}, W* = (0.57, 0.59, 0.58) dual stego image, I'''_{s1} dual stego image, I'''_{s2}

Figure 9. The visual inspection of the test image "Kodim20".



I_{sou}

I_{tar}

I''_{owct}, W* = (0.54, 0.49, 0.54) dual stego image, I'''_{s1} dual stego image, I'''_{s2}

Figure 10. The visual inspection of the test image "Kodim23".

Figures 3 and 4 exhibit two cases of stego images, "General Sherman" and "Clock Building," produced by our scheme. The target images are "Cemetery Tree (1)" and "Ahwahnee Great Lounge," and $\boldsymbol{W}^*$= (0.58, 0.60, 0.71) represents the optimal weight vector derived from the Kullback–Leibler divergence (KLD) in the Red, Green, and Blue channels [34]. Each pixel in the stego image, $\boldsymbol{I''_{owct}}$, is produced by an optimal weighted color-transfer and message embedding, which has concealed 3-ary message vector, $\boldsymbol{d_{sb}}$, and a 7-ary secret message, d_{wm}. Also shown are the dual stego images, I'''_{s1} and I'''_{s2}. Two pixels have concealed a 5-ary secret message vector, $\boldsymbol{d_{du}}$. The stego images do not show any artifact due to message concealment, even though the embedding rate is as high as 3.682 bits-per-pixel-per-stego image.

4.1. Embedding Capacity Analysis

In this section, we discuss the embedding capacity of our proposed method. In phase-1, we conceal an n-ary secret message vector, $\boldsymbol{d_{sb}}$, and an m-ary secret message, d_{wm}, in each pixel when synthesizing a stego image I''_{owct} using an optimal weighted color transfer. The embedding capacity (EC_1) in bits in Phase-1 is shown in Equation (15), where W×H represents the size of the stego image. Furthermore, the embedding rate (ER_1) in bits-per-pixel (bpp) is shown in Equation (16), which depends on two parameters, n and m, but is independent of the size of the stego image.

$$EC_1 = EC_1(SB) + EC_1(WM) = (\log_2 n^3) \times W \times H + (\log_2 m) \times W \times H \quad (15)$$

$$ER_1 = (\log_2 n + \log_2 m^{\frac{1}{3}}) \quad (16)$$

In phase-2, we adopt the weighted modulus reversible data hiding (WMRDH) for message concealment. We construct a 3-tuple dual-pixel, $\boldsymbol{P''_{du}}$, consisting of three component pairs, $\boldsymbol{P''_{r1}}$, $\boldsymbol{P''_{g2}}$, and $\boldsymbol{P''_{b3}}$, in the Red, Green, and Blue channel, respectively, which are constructed from the input stego image, I''_{owct}. Since each component pair can carry a k-ary secret message, two pixels, $\boldsymbol{P'''_{s1}}$ and $\boldsymbol{P'''_{s2}}$, in the respective dual stego images, I'''_{s1} and I'''_{s2}, have three component pairs, thereby able to convey a k-ary 3-tuple secret message vector, $\boldsymbol{d_{du}}$. Consequently, the embedding capacity carried in dual stego images is expressed in Equation (17), and the embedding rate (number of bits-per-stego pixel and per-stego image) is shown in Equation (18).

$$EC_2 = (3 \times \log_2 k) \times W \times H \quad (17)$$

$$ER_2 = \frac{1}{2} \log_2 k \quad (18)$$

Table 7 shows the embedding capacities produced from two phases. The first phase, Phase-1, conceals an n-ary secret message vector, $\mathbf{d_{sb}}$, denoted by $EC_1(SB)$ and m-ary secret message, d_{wm}, denoted by $EC_1(WM)$. The embedding capacity, EC_1, is the sum of $EC_1(SB)$ and $EC_1(WM)$. The second phase, Phase-2, embeds a k-ary secret message in a component pair, thereby offering the embedding capacity, EC_2(WMRDH), and the embedding rate, ER_2. In our experiments, we set n = 3, m = 7, and k = 5, implying that we embed 3-ary and 7-ary secret messages in Phase-1, producing the stego optimal weighted color-transferred image, I''_{owct}. We then convey 5-ary secret messages in Phase-2, generating the dual stego images I'''_{s1} and I'''_{s2}.

Table 7. The embedding capacities and embedding rates in Phase-1 (SB+WM) and Phase-2 (WMRDH).

No.	Source	EC_1(SB)	EC_1 (WM)	EC_1	ER_1	EC_2(WMRDH)	ER_2	EC (Total)	ER (Total)
1	Baboon	1,246,465	735,931	1,982,396	2.521	1,826,039	1.161	3,808,435	3.682
2	Lena	1,246,465	735,931	1,982,396	2.521	1,826,039	1.161	3,808,435	3.682
3	Pepper	1,246,465	735,931	1,982,396	2.521	1,826,039	1.161	3,808,435	3.682

4	Kodim07	1,869,697	1,103,896	2,973,593	2.521	2,739,058	1.161	5,712,651	3.682
5	Kodim20	1,869,697	1,10,3896	2,973,593	2.521	2,739,058	1.161	5,712,651	3.682
6	Kodim23	1,869,697	1,103,896	2,973,593	2.521	2,739,058	1.161	5,712,651	3.682

Therefore, our algorithm can provide total embedding capacity between 38.08~57.12 million bits, equivalent to the embedding rate of 3.682 bits-per-pixel-per-stego image.

Table 8 displays the timing results for six test images, including the optimal weight (W^*) in three channels, (a) time spent in determining the optimal weight, (b) timing of concurrently proceeding with the optimal weighted color-transfer and secret message embedding in Phase-1, (c) timing of reversible data-hiding weighted modulus in Phase-2. We also list the timing results for (d) the time required to extract secret message in Phase-2 and image restoration, and finally (e) the time spent in extracting secret messages in Phase-1. We can see from the statistics that the most time-consuming step is to determine the optimal weights which takes several seconds, which greatly depends on the size of the processing image. The time spent in all other processes is fast, less than one second.

Table 8. The timing results of six test images.

No.	Source	Target	Resolutions.	W*(R, G, B)	(a)	(b)	(c)	(d)	(e)
1	Baboon	Barbara	512 × 512	(0.45, 0.48, 0.50)	9.554	0.318	0.126	0.039	0.02
2	Lena	Goldhill	512 × 512	(0.48, 0.60, 0.42)	9.523	0.298	0.125	0.053	0.02
3	Pepper	Boats	512 × 512	(0.47, 0.54, 0.48)	9.352	0.317	0.119	0.035	0.018
4	Kodim07	Kodim24	512 × 768	(0.48, 0.46, 0.48)	14.235	0.516	0.203	0.066	0.03
5	Kodim20	Kodim22	512 × 768	(0.57, 0.59, 0.58)	13.933	0.454	0.187	0.056	0.03
6	Kodim23	Kodim11	512 × 768	(0.54, 0.49, 0.54)	13.616	0.445	0.191	0.07	0.032

4.2. Image Quality Analysis

In this subsection, we adopt ENIQA [35] and BRISQUE [36] to evaluate the image quality of the stego image, I''_{owct}, generated in Phase-1. ENIQA is a high-performance, general-purpose, no-reference assessment method based on image entropy. The range of the ENIQA score is within [0, 1]; a lower score represents higher quality. BRISQU is another high-performance, general-purpose, no-reference assessment method based on the Natural Scene Statistics (NSS) extraction and the feature vectors calculation. The BRISQUE score is usually in the range [0, 100]; smaller scores reflect better perceptual quality of the test image with respect to the input model.

Table 9 shows the ENIQA and BRISQUE scores for the source, target, and the stego image, I''_{owct}, where its colors are borrowed from those in the source and target images. using an optimal weight vector. The average ENIQA scores is 0.0975, which is close to 0, while the average BRISQUE score is 13.9201, which is far from the worse scores, 100. Both scores reflect that our scheme can produce a stego weighted color-transferred image that exhibits good image quality.

Table 9. No-reference image quality assessment for 6 test cases.

NO.	Source		Target		I''_{owct}	
	ENIQA	BRISQUE	ENIQA	BRISQUE	ENIQA	BRISQUE
1	0.2157	16.1316	0.0814	35.7050	0.2125	17.2771
2	0.0952	2.4450	0.1726	12.4404	0.0755	3.1437
3	0.1233	19.1092	0.3492	42.8666	0.1312	21.3583
4	0.1028	16.9239	0.0219	13.6321	0.0961	22.7102
5	0.0102	1.8924	0.0099	11.3557	0.0285	7.6039
6	0.0120	12.4011	0.0222	7.9306	0.0414	11.4272
Avg.	0.0932	11.4839	0.1095	20.6551	0.0975	13.9201

Figures 5–10 show the visual inspection of six test images along with the optimal weighted vectors, W*. We exhibit the source, target image, the stego optimal weighted color transferred image, I''_{owct}, and the final dual stego images: I'''_{s1} and I'''_{s2}.

Table 10 exhibits the results of the full-reference image quality assessment, where we consider the optimal weighted color-transferred image, I''_{owct}, as the reference image and the dual stego images, I'''_{s1} and I'''_{s2}, as the test images. We list the other metrics of the full-reference image quality assessment in Table 11. Specifically, we report the following standard metrics to evaluate the image quality: Mean Square Error, Peak Signal Noise Ratio (PSNR) [37], Visual Saliency-induced Index (VSI) [38], Structural Similarity Index (SSIM) [39], Information Content Weighting Mean Squared Error (IW-MSE) [40], University Image Quality Index (Q-Index) [41,42]. In metrics introduced by [38,39,41], scores are within the range [0, 1]; the score close to 1.0 represents that the test image has less distortion in comparison with the reference image.

Table 10. Full-reference image quality assessment for six test images (MSE, IW-MSE, PSNR).

No.	MSE		IW-MSE		PSNR	
	I'''_{s1}	I'''_{s2}	I'''_{s1}	I'''_{s2}	I'''_{s1}	I'''_{s2}
1	0.6003	0.6001	0.15196	0.14961	50.3472	50.3484
2	0.6000	0.5995	0.15062	0.15014	50.3496	50.3532
3	0.6003	0.5995	0.14839	0.15128	50.3468	50.3528
4	0.6009	0.5998	0.15121	0.15028	50.3425	50.3505
5	0.5996	0.5994	0.15016	0.14969	50.3519	50.3533
6	0.6007	0.5996	0.14839	0.14867	50.3441	50.3523
Avg.	0.6003	0.5997	0.15012	0.14995	50.3470	50.3518

Table 11. Full-reference image quality assessment for six test stego images.

No.	SSIM		Q-Index		VSI		IW-SSIM	
	I'''_{s1}	I'''_{s2}	I'''_{s1}	I'''_{s2}	I'''_{s1}	I'''_{s2}	I'''_{s1}	I'''_{s2}
1	0.99974	0.99974	0.999996	0.999996	0.99997	0.99997	0.99988	0.99988
2	0.99936	0.99936	0.999991	0.999990	0.99996	0.99996	0.99967	0.99967
3	0.99932	0.99932	0.999995	0.999996	0.99995	0.99995	0.99965	0.99965
4	0.99938	0.99940	0.999996	0.999996	0.99996	0.99996	0.99974	0.99974
5	0.99886	0.99886	0.999993	0.999993	0.99994	0.99994	0.99952	0.99952
6	0.99899	0.99900	0.999993	0.999993	0.99995	0.99995	0.99948	0.99948
Avg.	0.99927	0.99928	0.999994	0.999994	0.99995	0.99995	0.99966	0.99966

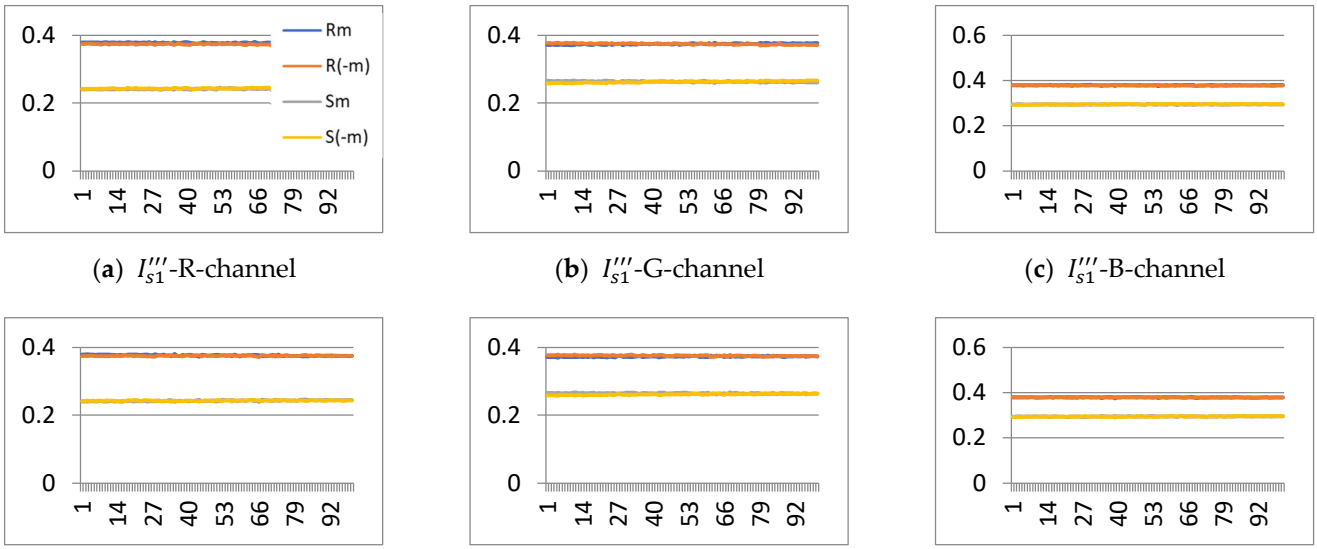
The values for MSE or IW-MSE are small, less than 0.61 and no more than 0.15, respectively (the smaller, the better). In contrast, the PSNR values are high, over 50.34 dB (the higher, the better). Finally, scores for SSIM, Q-index, VSI, and IW-SSIM are close to 1.0. All the statistics indicate that there is no notable difference between the reference and the test images. These results confirm that the distortion in the dual stego images is insignificant, even though they have concealed a large number of secret messages.

4.3. RS Steganalysis Analysis

Steganalysis aims to detect any messages conveyed in a stego image. The statistical hypothesis of the well-known RS steganalysis method [43] is that in a typical image with no hidden message, the expected value of R_m equals that of R_{-m} (the regular groups with masks, m and -m), and the same is true for S_m and S_{-m} (the singular groups with masks m and -m).

Figure 11 shows the RS diagram for the RGB-channel of the dual stego images produced by our algorithm. The diagram depicts regular and singular groups with respect to different masks as functions of the embedding ratio from 1 to 100 %. As shown in the

graph, $R_m > S_m$, $R_{-m} > S_{-m}$, $R_m \cong R_{-m}$, and $S_m \cong S_{-m}$ for all embedding ratios, indicating that the difference between R_m and R_{-m} exhibits no significant change as the number of embedded messages increases. In other words, the message embedding does not substantially change the number of regular or singular groups. These figures confirm that our proposed scheme is able to resist the RS steganalysis attack.



(**a**) I'''_{s1}-R-channel (**b**) I'''_{s1}-G-channel (**c**) I'''_{s1}-B-channel

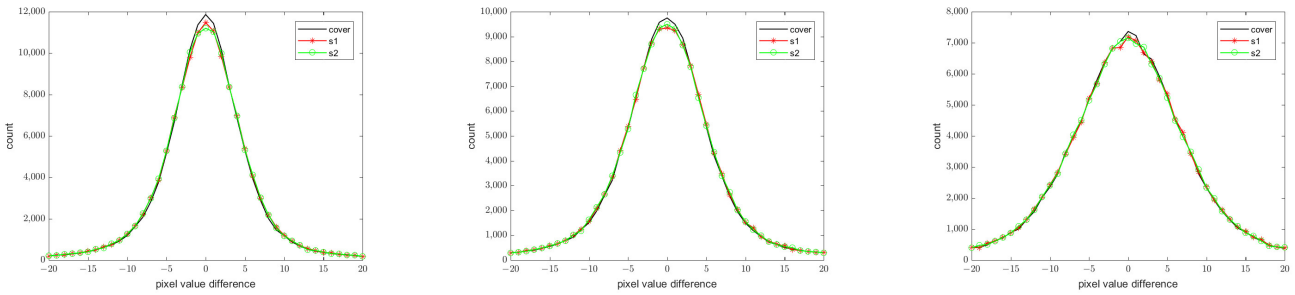
(**d**) I'''_{s2}-R-channel (**e**) I'''_{s2}-G-channel (**f**) I'''_{s2}-B-channel

Figure 11. The RS diagrams for dual "Lena" (top row) and "Goldhill" (bottom row) stego images.

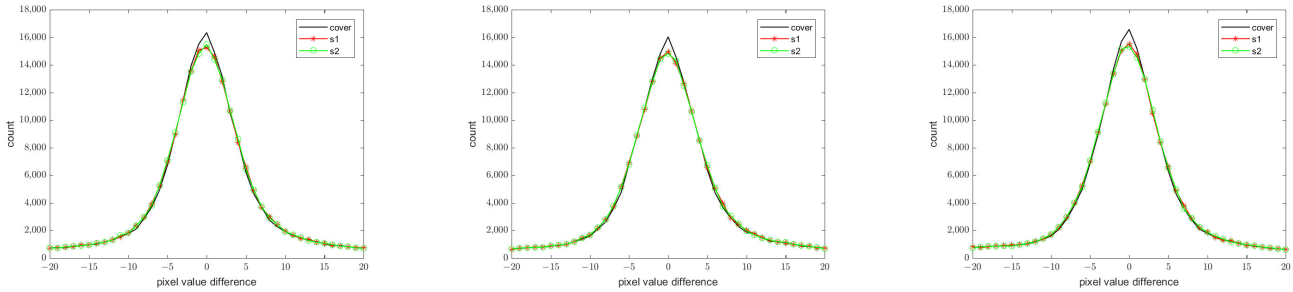
4.4. PVD Analysis

The difference between two neighboring pixels represents the smoothness of an image, which can be displayed via the pixel value differences (PVD) histogram. Since the message embedding will affect the pixel values, one can examine the shape of the PVD histogram to reveal any secret messages hidden in an image for steganalysis.

Figure 12 shows the PVD histogram in I''_{owct}, which is considered as the cover, and the PVD histogram in the dual stego images, I'''_{s1} and I'''_{s2}, are denoted as "s1" and "s2," respectively. The diagrams of three PVD histograms closely overlap, indicating that there is no substantial change in the shape of the PVD histograms before and after the message concealment. Thus, our algorithm preserves the smoothness in I'''_{s1} and I'''_{s2}, and its ability to resist the PVD histogram attack.



(**a**) R-channel (**b**) G-channel (**c**) B-channel



(**d**) R-channel (**e**) G-channel (**f**) B-channel

Figure 12. The PVD histograms for the test image "Lena" (top row) and those for "Kodim07".

4.5. Statistical Analysis

Statistical analysis is a conventional method used in steganalysis. The idea is analyzing the statistical changes after message embedding and measuring changes by statistical metrics, such as standard deviation and correlation coefficients in three directions: horizontal, vertical, and diagonal. If the correlation coefficients are close to 1.0 after the message embedding, there is less pixel distortion, thereby, with a greater ability to resist the statistical steganalytic attacks.

Table 12 shows the correlation coefficients for I''_{owct}, I'''_{s1}, and I'''_{s2} in three directions in the Red channel. The coefficients are large; some are close to 1.0. In addition, the correlation coefficients are similar, with the divergence being less than 0.0001. Table 13 presents the standard deviation for the test images. Their divergences are not significant, with the difference less than 0.01. Table 14 shows the correlation coefficients in three chromatic channels, which are very close to 1. These statistics demonstrate the ability of our algorithm to resist the statistical steganalytic attacks.

Table 12. The correlation coefficients of six test images in three directions in Red channel.

NO	I''_{owct}			I'''_{s1}			I'''_{s2}		
	Diag.	**Vert.**	**Hor.**	**Diag.**	**Vert.**	**Hor.**	**Diag.**	**Vert.**	**Hor.**
1	0.8525	0.8642	0.9215	0.8523	0.8640	0.9213	0.8524	0.8641	0.9214
2	0.9699	0.9894	0.9805	0.9697	0.9891	0.9803	0.9697	0.9892	0.9803
3	0.9552	0.9651	0.9623	0.9550	0.9649	0.9621	0.9549	0.9649	0.9621
4	0.9336	0.9545	0.9675	0.9333	0.9543	0.9672	0.9334	0.9543	0.9672
5	0.9800	0.9858	0.9895	0.9798	0.9857	0.9894	0.9798	0.9857	0.9894
6	0.9781	0.9832	0.9876	0.9779	0.9830	0.9873	0.9779	0.9830	0.9873

Table 13. The standard deviation of the six test images in three chromatic channels.

No.	I''_{owct}			I'''_{s1}			I'''_{s2}		
	R	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**
1	55.31	48.37	58.08	55.31	48.37	58.09	55.31	48.37	58.09
2	52.87	53.84	45.61	52.87	53.85	45.61	52.87	53.85	45.61
3	50.37	64.62	50.43	50.38	64.62	50.44	50.38	64.62	50.44
4	46.72	45.27	45.57	46.73	45.28	45.57	46.72	45.28	45.57
5	62.52	59.93	65.25	62.52	59.93	65.26	62.52	59.93	65.26
6	47.98	48.09	47.93	47.99	48.10	47.94	47.99	48.10	47.94

Table 14. The correlation coefficients of dual stego images in the Red, Green, and Blue channels.

No.	I'''_{s1}			I'''_{s2}		
	R	G	B	R	G	B
1	0.99991	0.99988	0.99992	0.99991	0.99988	0.99992
2	0.99990	0.99990	0.99987	0.99990	0.99990	0.99987
3	0.99989	0.99993	0.99989	0.99989	0.99993	0.99989
4	0.99987	0.99986	0.99986	0.99987	0.99986	0.99987
5	0.99993	0.99992	0.99993	0.99993	0.99992	0.99993
6	0.99988	0.99988	0.99988	0.99988	0.99988	0.99988

4.6. Robustness

This section presents the robustness evaluation of our proposed method when the stego images are under four types of attacks: 1% and 5% of the salt-and-pepper (s&p) noise attacks at each channel and the cropping attack under two different image sizes (96 × 96 or 128 × 128 pixels). In the salt-and-pepper noise attack, around 1% and 5% of channels have been set to black or white, while pixels within areas under the cropping attack are set to black.

Table 15 shows the bit error rates (BER) derived from four test images when extracting secret bits from stego images under the salt-and-pepper or cropping attacks. Zero BER indicates that all extracted bits are identical to the embedding ones; thus, the smaller the BER, the better the robustness. The statistics show that when the stego images are under s&p attacks, the BERs are small when concealing messages using the single base (SB) approach, but they are larger if we conceal messages using the weighted modulus (WM). These results are reasonable because the message embedding scheme, as shown in Equation (6), is vulnerable to any changes in the Red, Green, or Blue channels due to the fact that messages must be concealed via the vector dot operation. In contrast, the SB scheme, as shown in Equation (7), is more robust to any changes in the Red, Green, or Blue channels, as messages are concealed independently in each channel. Nevertheless, the overall results for the BER evaluation are satisfactory.

Table 15. The bit error rate and image quality under the salt-and-pepper and crop attacks.

Image	Type	BER(DUAL)	BER(SB)	BER(WM)	PSNR	VSI	SSIM	Q-Index
Kodim07	S&P (1%)	1.588%	1.312%	4.995%	27.23	0.99211	0.94924	0.99926
Kodim23	S&P (5%)	7.803%	6.482%	22.615%	20.85	0.95783	0.92022	0.99713
Baboon	Cropping (96)	5.752%	4.789%	6.143%	22.73	0.99223	0.96217	0.97417
Lena	Cropping (128)	10.188%	8.532%	10.837%	21.62	0.98961	0.94171	0.95237

Table 15 also exhibits the image quality results. The reference image is the restored image derived from a stego image *without* any attacks, while the test image is the restored image from the one under various types of attacks. The PSNR values are greater than 20 dB; the lower the attack ratios, the larger the PSNR values. In addition, the experimental results, revealing a high SSIM, larger than 0.941, a high Q-index, larger than 0.952, indicates that the restored image continues to possess high image quality, even when the stego images are under various attacks. The experimental results confirm that our scheme is robust, able to resist the salt-and-pepper noise and image cropping attacks.

Figure 13 exhibits visual inspection of the attacked stego images. Note that the attacked areas in the restored image, RI''_{owct}, are not all black because pixel values are still correct in some of the Red, Green, and Blue channels. Nevertheless, the details of the

image contents can still remain visible, and the effects of attacks are limited, thereby indicating the ability to resist the noise and cropping attacks.

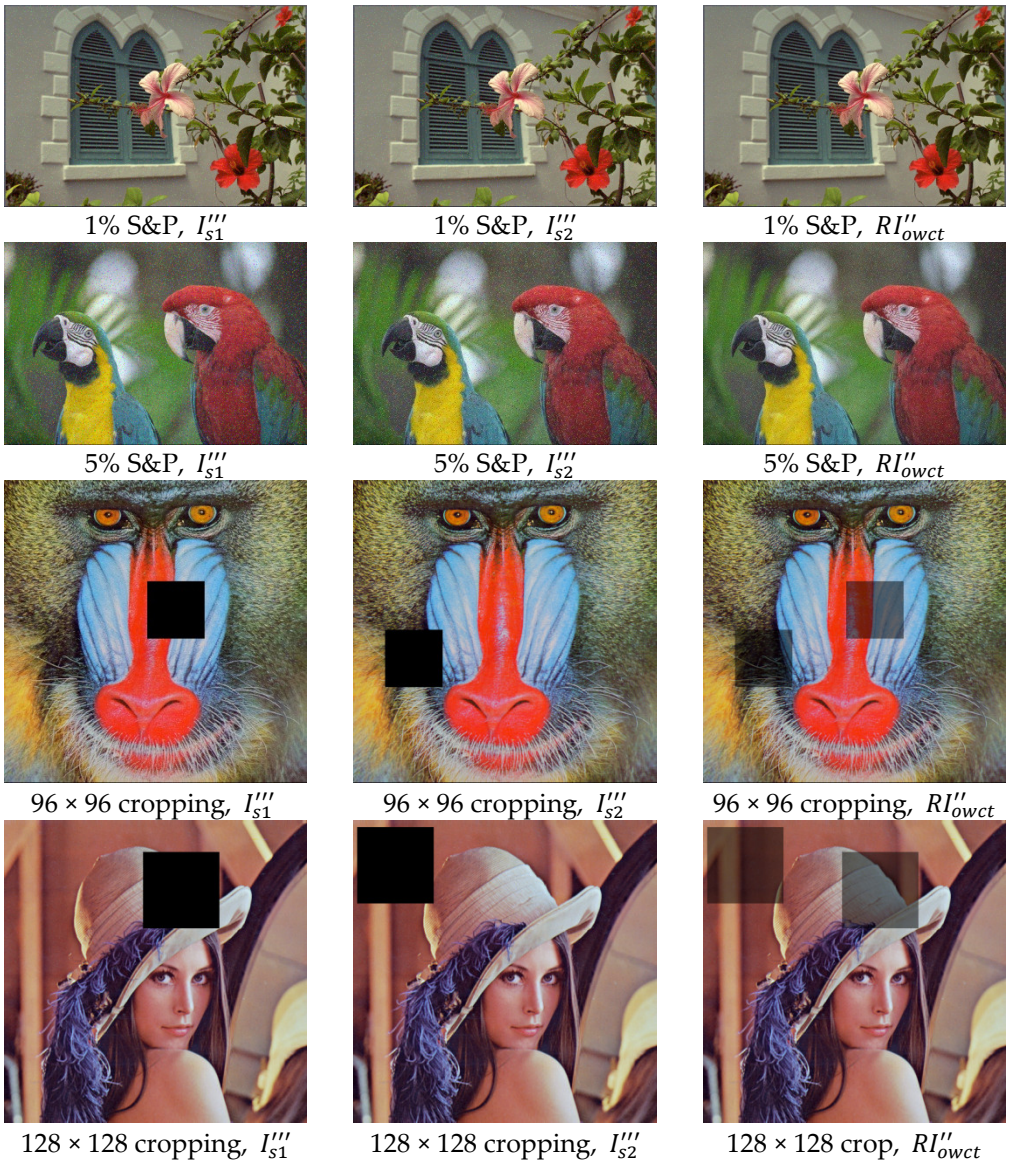


Figure 13. The visual inspection results of four types of attack and the corresponding restored image.

4.7. Comparison with the Current State-of-the-Art Works

In this section, we compare the current state-of-the-art works with our proposed scheme in Phase-2, weighted modulus reversible data hiding (WMRDH), which conceals the three secret messages. The comparison is due to the fact that most of our competitors embed a single level of secret message. To provide a fair comparison, we set k=5 in our experiment, indicating that we conceal a 5-ary secret message to produce dual stego images. We compare our scheme with previous works, including Chang et al. [18], Qin et al. [20], Lu et al. [22], Lin et al. [26], Chen and Chang [30], and Xie et al. [31].

Table 16 shows the average PSNR values of each test image (marked with "*"), the embedding ratio in bpp, and the embedding efficiency. Note that the embedding efficiency (EE) represents the embedding ratio (bpp) over the image distortion in mean square error (MSE), thereby signifying the overall message-embedding performance, i.e., the higher, the better. The embedding ratio of the proposed method is about 1.161 bpp, the average PSNR (Peak Signal-to-Noise Ratio) is 50.35 dB, and the embedding efficiency

is 1.935. Chang et al. [18] have a high embedding ratio, but small PSNR values and small embedding efficiency. This means that their method conceals a large number of messages but produces significant image distortion as well. Our algorithm has higher embedding efficiency, thus outperforming Chang et al.'s.

Qin et al.'s [20] embedding ratio is similar to our method, but ours has much higher PSNR (e.g., 46.84 vs 50.35 in the Lena test image), thereby indicating less image distortion. Although the average of PSNR in [20] is about 46.85 dB, the dual stego images generated have significantly different PSNR values.

The embedding ratio of Lu et al. [22] is 1.00 bpp, and the average PSNR is 49.10 dB. The statistics show that our scheme performs better than [22], both on the embedding capacity and the image quality. The embedding ratio of Jafar et al. [24] is 1.2413 bpp, and the average PSNR is 48.71 dB. The statistics show that our scheme performs better than their schemes in the aspect of the embedding efficiency. In addition, our algorithm offers much larger embedding efficiency than Jana et al.'s work [25]. The embedding ratio of Lin et al. [26] is higher than ours, but the average PSNR is 47.46 dB. Thus, the comparison in terms of the embedding efficiency confirms that our method is better than [26].

The embedding ratio of the proposed method is close to Chen and Chang [30], but our scheme has higher PSNR (e.g., 48.72 vs. 50.35 in the Lena test image). The average PSNR value of [30] is 48.72 dB, lower than ours (50.35 dB). Finally, the embedding ratio of Xie et al. [31] is higher than ours, but the average PSNR just reaches 43.04 dB, much lower than that of our scheme. The embedding efficiency demonstrates our scheme is superior to [31]. We conclude that the overall comparison confirms that our scheme outperforms all eight competitors.

Table 16. Comparison results in terms of PSNR, embedding rate, and embedding efficiency.

Images	Method	Proposed	[18]	[20]	[22]	[24]	[25]	[26]	[30]	[31]
Barbara	PSNR* (dB)	50.348	39.89	46.85	49.12	48.71	54.27	47.47	48.72	48.72
	ER (bpp)	1.161	1.5314	1.16	1.00	1.2413	0.1867	1.25	1.1274	1.1274
	EE	1.935	0.2296	0.8637	1.2558	1.4167	0.7676	1.0736	1.2912	1.2912
Boat	PSNR* (dB)	50.352	39.89	46.84	49.03	48.71	-	47.46	48.72	43.04
	ER (bpp)	1.161	1.531	1.16	1.00	1.2419	-	1.25	1.1271	1.6666
	EE	1.935	0.2296	0.8617	1.23	1.4174	-	1.0711	1.2909	0.5161
Goldhill	PSNR* (dB)	50.353	39.90	46.85	49.13	48.72	53.73	47.46	48.72	43.03
	ER (bpp)	1.161	1.531	1.16	1.00	1.2412	0.1867	1.25	1.1274	1.6659
	EE	1.935	0.2301	0.8637	1.2587	1.4199	0.6778	1.0711	1.2912	0.5147
Lena	PSNR* (dB)	50.351	39.89	46.84	49.13	48.71	53.96	47.47	48.72	43.04
	ER (bpp)	1.161	1.5314	1.16	1.00	1.2405	0.1867	1.25	1.1274	1.6676
	EE	1.935	0.2296	0.8617	1.2587	1.4158	0.7147	1.0736	1.2912	0.5164
Pepper	PSNR* (dB)	50.349	39.94	46.42	49.09	48.71	53.94	47.46	48.72	43.03
	ER (bpp)	1.161	1.5253	1.16	1.00	1.1971	0.1867	1.25	1.1273	1.6673
	EE	1.935	0.2314	0.7823	1.2472	1.3679	0.7114	1.0711	1.2911	0.5151

5. Conclusions

In this paper, we introduce a novel multi-hider reversible data-hiding scheme. Our algorithm presents the advantage of a multi-hider approach; we concurrently synthesize an image and conceal the first secret message vector in the first phase, before conveying the second secret message followed by the third secret message vector in the second phase. The message concealment within two phases does not affect the messages already conveyed. Consequently, our algorithm is able to independently embed three types of secret message through a reversible data-hiding approach, thereby producing dual stego images. According to dissimilar secret keys, legitimate recipients can extract distinct secret messages from seven levels of message extraction to accomplish a secure data communication.

Security analysis confirms that our scheme is statistically secure, able to resist the RS steganalysis and the PVD steganalytic attacks. The experimental results show that the proposed method achieves the highest embedding capacity and produces high quality stego images, outperforming those of our competitors. To the best of our knowledge, our scheme is the first in the relevant literature that provides multi-hider reversible data hiding, offering various levels of message extraction for secure data communication.

Author Contributions: K.-S.H.: conceptualization, methodology, software, validation, formal analysis, investigation, data curation, visualization. C.-M.W.: conceptualization, methodology, validation, formal analysis, investigation, writing—original draft, writing—review. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: This work is supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2221-E-005-069, 108-2221-E-005-051, MOST 109-2221-E-005-062, MOST 110-2221-E-005-069, and NSC-111-2221-E-005 -076. The authors acknowledge the comments and feedback they received from the anonymous reviewers, all of which helped to improve the presentation of the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kadhim, I.J.; Premaratne, P.; Vial, P.J.; Halloran, B. Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. *Neurocomputing* **2019**, *335*, 299–326, 10.1016/j.neucom.2018.06.075.
2. Chan, C.K.; Chen, L.M. Hiding data in images by simple LSB substitution. *Pattern Recognit.* **2004**, *37*, 469–474.
3. Fridrich, J.; Soukal, D. Matrix embedding for large payloads. *IEEE Trans. Inf. Forensics Secur.* **2006**, *1*, 390–395.
4. Zhang, X.; and Wang, S. Efficient steganographic embedding by exploiting modification direction. *IEEE Commun. Lett.* **2006**, *10*, 781–783.
5. Ker, A. Improved detection of LSB steganography in grayscale images. In *International Workshop on Information Hiding*; Springer: Berlin, Heidelberg, 2004; pp. 97–115.
6. Mao, Q. A fast algorithm for matrix embedding steganography. *Digit. Signal Process.* **2014**, *25*, 248–254.
7. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896.
8. Ni, Z.C.; Shi, Y.Q.; Ansari, N.; Su, W. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–363.
9. Li, X.; Li, J.; Li, B.; Yang, B. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Process*. **2013**, *93*, 198–205.
10. Alattar, A.M. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans. Image Process.* **2004**, *13*, 1147–1156.
11. Al-Qershi, O.M.; Khoo, B.E. Two-dimensional difference expansion (2D-DE) scheme with a characteristics-based threshold. *Signal Process*. **2013**, *93*, 154–162.
12. Tsai, P.; Hu, Y.C.; Yeh, H.L. Reversible image hiding scheme using predictive coding and histogram shifting. *Signal Process*. **2009**, *89*, 1129–1143.
13. Tai, W.L.; Yeh, C.M.; Chang, C.C. Reversible data hiding based on histogram modification of pixel differences. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 906–910.
14. He, W.; Zhou, K.; Cai, J.; Wang, L.; Xiong, G. Reversible data hiding using multi-pass pixel value ordering and prediction-error expansion. *J. Vis. Commun. Image Represent.* **2017**, *49*, 351–360.
15. Peng, F.; Li, X.L.; Yang, B. Improved PVO-based reversible data hiding. *Digit. Signal Process.* **2014**, *25*, 255–265.
16. Chang, C.C.; Kieu, T.D.; Chou, Y.C. Reversible data hiding scheme using two steganographic images. In Proceedings of the TENCON 2007–2007 IEEE Region 10 Conference, Taipei, Taiwan, 30 October–2 November 2007; pp. 1–4.
17. Chang, C.C.; Chou, Y.C.; Kieu, T.D. Information hiding in dual images with reversibility. In Proceedings of the 2009 Third International Conference on Multimedia and Ubiquitous Engineering, Qingdao, China, 4–6 June 2009; pp. 145–152.
18. Chang, C.C.; Lu, T.C.; Horng, G.; Huang, Y.H.; Hsu, Y.M. A high payload data embedding scheme using dual stego-images with reversibility. In Proceedings of the 2013 9th International Conference on Information, Communications & Signal Processing, Tainan, Taiwan, 10–13 December 2013; pp. 1–5.

19. Lee, C.F.; Huang, Y.L. Reversible data hiding scheme based on dual stegano-images using orientation combinations. *Telecommun. Syst.* **2013**, *52*, 2237–2247.
20. Qin, C.; Chang, C.C.; Hsu, T.J. Reversible data hiding scheme based on exploiting modification direction with two steganographic images. *Multimed. Tools Appl.* **2015**, *74*, 5861–5872.
21. Lu, T.C.; Wu, J.H.; Huang, C.C. Dual-image-based reversible data hiding method using center folding strategy. *Signal Process.* **2015**, *115*, 195–213.
22. Lu, T.C.; Tseng, C.Y.; Wu, J.H. Dual imaging-based reversible hiding technique using LSB matching. *Signal Process.* **2015**, *108*, 77–89.
23. Yao, H.; Qin, C.; Tang, Z.; Tian, Y. Improved dual-image reversible data hiding method using the selection strategy of shiftable pixels' coordinates with minimum distortion. *Signal Process.* **2017**, *135*, 26–35.
24. Jafar, I.F.; Darabkh, K.A.; Al-Zubi, R.T.; Saifan, R.R. An efficient reversible data hiding algorithm using two steganographic images. *Signal Process.* **2016**, *128*, 98–109.
25. Jana, B.; Giri, D.; Mondal, S.K. Dual image based reversible data hiding scheme using (7, 4) hamming code. *Multimed. Tools Appl.* **2018**, *77*, 763–785.
26. Lin, J.Y.; Liu, Y.; Chang, C.C. A real-time dual-image-based reversible data hiding scheme using turtle shells. *J. Real-Time Image Process.* **2019**, *16*, 673–684.
27. Shastri, S.; Thanikaiselvan, V. Dual image reversible data hiding using trinary assignment and centre folding strategy with low distortion. *J. Vis. Commun. Image Represent.* **2019**, *61*, 130–140.
28. Yao, H.; Mao, F.; Tang, Z.; Qin, C. High-fidelity dual-image reversible data hiding via prediction-error shift. *Signal Process.* **2020**, *170*, 107447.
29. Lu, T.C.; Chang, T.C.; Shen, J.J. An effective maximum distortion controlling technology in the dual-image-based reversible data hiding scheme. *IEEE Access* **2020**, *8*, 90824–90837.
30. Chen, S.; Chang, C.C. Reversible data hiding based on three shadow images using rhombus magic matrix. *J. Vis. Commun. Image Represent.* **2021**, *76*, 103064.
31. Xie, X.Z.; Chang, C.C.; Lin, C.C. A hybrid reversible data hiding for multiple images with high embedding capacity. *IEEE Access* **2020**, *8*, 37–52. https://doi.org/10.1109/ACCESS.2019.2961764.
32. Chen, W.S.; Wu, K.C.; Wang, C.M. A novel message embedding algorithm using the optimal weighted modulus. *Inf. Sci.* **2017**, *388–389*, 17–36.
33. Hsieh, K.S.; Wang, C.M. Constructive image steganography using example-based weighted color transfer. *J. Inf. Secur. Appl.* **2022**, *65*, 103126.
34. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86.
35. Chen, X.; Zhang, Q.; Lin, M.; Yang, G.; He, C. No-reference color image quality assessment: From entropy to perceptual quality. *EURASIP J. Image Video* **2019**, 77.
36. Mittal, A.; Moorthy, A.K.; Bovik, A.C. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.* **2012**, *21*, 4695–4708.
37. Wang, Z.; Bovik, A.C. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Process. Mag.* **2009**, *26*, 98–117.
38. Zhang, L.; Shen, Y.; Li, H. VSI: A visual saliency-induced index for perceptual image quality assessment. *IEEE Trans. Image Process.* **2014**, *21*, 4270–4271.
39. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612.
40. Wang, Z.; Li, Q. Information content weighting for perceptual image quality assessment. *IEEE Trans. Image Process.* **2011**, *20*, 1185–1198.
41. Wang, Z.; Bovik, A.C. A universal image quality index. *IEEE Signal Process. Lett.* **2002**, *9*, 81–84.
42. Reinhard, E.; Ward, G.; Pattanaik, S.; Debevec, P.; Heidrich, W.; Myszkowski, K. *High Dynamic Range Imaging, Acquisition, Display, and Image-Based Lighting*, 2nd ed.; Morgan Kaufmann: Burlington, NJ, USA, 2010.
43. Fridrich, J.; Goljan, M.; Du, R. Detecting LSB steganography in color, and gray-scale images. *IEEE MultiMedia* **2001**, *8*, 22–28.