applied sciences

MDPI

Article

A Novel Adaptive Image Data Hiding and Encryption Scheme Using Constructive Image Abstraction

Chi-Feng Lan, Chung-Ming Wang * and Woei Lin

Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan; cflan@nchu.edu.tw (C.-F.L.); wlin@cs.nchu.edu.tw (W.L.)
* Correspondence: cmwang@cs.nchu.edu.tw

Abstract: Image abstraction simplifies complex images, highlights specific features, and preserves different levels of structures to achieve a desired style. This paper presents a constructive and adjustable data hiding algorithm to convey various secret messages and resist modern steganalytic attacks. Our scheme produces an abstracted stego image, while synthesizing an original image during the image abstraction process. Our algorithm is flexible, applicable to two types of images: high-dynamic-range images and ordinary color images, aka low-dynamic-range images. Additionally, we introduce a novel image encryption scheme suitable for the above two types of images, which incorporates a two-dimensional logistic tent modular map and a bit-level random permutation technique, thereby further protecting the content of the stego image and the carried secret messages. Compared with the current state-of-the-art methods, our algorithm provides a 14% to 33% larger embedding rate, while lowering the distortion of the abstracted stego image. A comprehensive security analysis confirmed that our algorithm provides high security to resist statistical, differential, brute force, chosen-plaintext, and chosen key attacks.

Keywords: constructive data hiding; high-dynamic-range image; message embedding; adaptivity; image abstraction; image encryption; information security

Citation: Lan, C.-F.; Wang, C.-M.; Lin, W. A Novel Adaptive Image Data Hiding and Encryption Scheme Using Constructive Image Abstraction. *Appl. Sci.* **2023**, *13*, 6208. https://doi.org/10.3390/app13106208

Academic Editors: David Megías, Minoru Kuribayashi and Wojciech Mazurczyk

Received: 13 April 2023
Revised: 8 May 2023
Accepted: 15 May 2023
Published: 18 May 2023

1. Introduction

Image abstraction transforms an input image into a new image with a distinct style while preserving the contents in the original image. The goal of image abstraction is to create visually appealing images with specific aesthetics, such as impressionism, cubism, or surrealism. This technique is useful to make images more visually appealing for social media or advertising. Winnemöller et al. [1] introduced a framework for automated and real-time video abstraction which employs Gaussian filters and bilateral filters. Ma et al. [2] proposed a quadruple-cycle framework to support iterative learning to achieve restorable arbitrary style transfer. A new quantitative evaluation method was introduced to measure content preservation and style embedding performance, while solving the content leak problem from an image restoration perspective. However, these methods consider the input image as a color image, also known as low-dynamic-range (LDR) image, as the pixel values are within the fixed range of [0, 1], before scaling up to [0, 255] for storage purpose. Consequently, these image abstraction algorithms are not applicable to high-dynamic-range (HDR) images which usually exceed the fixed range.

HDR images [3,4] have become increasingly popular due to their ability to offer a wider dynamic range. Dynamic range refers to the range of brightness or luminance values that are present in the image. HDR images offer improved image detail and a closer approximation to human visual perception compared to traditional LDR images. RGBE [5], LogLuv including LogLuv24 and LogLuv32 [6], and OpenEXR [7] are three major established HDR image file formats. In contrast to the integer representation in LDR images, HDR images are able to precisely represent a broad range of luminance and colors

by utilizing floating-point numbers to store critical information. This unique capability positions HDR images as a promising candidate to become the dominant image standard in the future. In particular, high-dynamic-range imaging [8] represents a technique for accurately representing a wide range of intensity levels captured in real scenes from an extensive dynamic range of exposures; it has been intensively investigated with applications in image processing, computer graphics, and computer vision. In contrast to high-dynamic-range imaging, only a limited number of HDR image abstraction techniques have been proposed. Zhou et al. [9] modified the image filtering scheme of [1] to solve the image abstraction transfer problem and successfully abstracted HDR images. Kumar et al. [10] introduced an integrated filter-based approach to produce the effect of line drawing from HDR images; their scheme incorporates the features of anisotropic filter, shock filter with iterations, and bilateral filter to produce the important shapes in HDR images without upsetting the human visual effect.

Regarding data hiding for HDR images, several studies proposed an adaptive data hiding approach to balance image quality and embedding rate. These approaches leverage the fact that HDR images often have a wide range of luminance and colors, while the human eye has different sensitivity in different luminance and texture. Cheng and Wang [11] pioneered these works and pixels in RGBE images are categorized into homogeneous or heterogeneous areas to carry adjustable secret data. They offered an embedding rate ranging from 5.13 to 9.69 bits per pixel (bpp); however, the peak signal-to-noise ratios (PSNR) of the resultant tone-mapped stego images were only slightly greater than 30 dB. By exploiting the homogeneous representations of RGBE pixels, Yu et al. [12] concealed different numbers of messages based on the homogeneous representations and achieved data hiding without distortion in HDR RGBE images, but their scheme has to pay the penalty of providing a limited embedding rate in the range of 0.127–0.145 bpp. Gao et al. [13] utilized a two-dimensional prediction-error histogram to decrease the distortion from data hiding and provide between 1.202 and 2.85 embedding rates in bpp. Lan et al. [14] introduced an adaptive method to carry secret message in RGBE images, which takes into consideration of the exponent channel distribution when converting from OpenEXR image format. Their method keeps the exponent channel intact during the message concealment. Their scheme supports embedding rates from 7.30 to 9.29 bpp. Lin et al. [15] partitioned the pixels into three groups according to their luminance, and selected low luminance pixels to conceal a greater number of messages. Depending on the specific parameters used, their approach yielded the embedding rates of 2.433 to 20.002 bpp.

Most of the above-mentioned works embed messages by altering pixels of the cover image, thereby generating stego pixels with different degrees of distortion, depending on the number of secret messages injected into the cover images. In this scenario, the format remains consistent between the cover and stego images. As a result, concealing a greater message results in a larger image distortion, making it challenging to withstand steganalytic attacks, which aim to identify any hidden messages in an innocent-looking image. To address the issue of possible steganalytic vulnerabilities, researchers have proposed a novel data hiding scheme that conceals confidential messages by creating a stego image directly, instead of modifying an existing cover image. This method is known as "constructive data hiding" or "constructive steganography" [16]. Hsieh and Wang [17], evaluated the EC-MV steganalysis of their WCTCIS algorithm using the BOSSbase2 image data base containing 10,000 color images. Their algorithm, which uses the constructive image steganography (CIS) approach, can resist the steganalysis attack. Their experimental results demonstrate that the detection rate in the receiver operating characteristic (ROC) curve is linear and the area under the curve (AUC) is around 0.50, which is no better than a random guess. They also reported that in contrast, the steganographic algorithm by cover modification (SCM) approach has a high detection rate, resulting in the AUC being as high as 0.9984. Hsieh and Wang concluded that the stego images produced by the CIS approach can resist modern power attacks, such as EC-MV steganalysis, RS steganalysis, pixel value difference (PVD) attacks, and an attack through statistical analysis.

Similar to data hiding methods, encryption algorithms for HDR images are limited in the literature. Some techniques have been proposed to encrypt different aspects of HDR images. Yan et al. [18] applied elementary cellular automata (ECA) to encrypt HDR RGBE images, while Lin et al. [19] generated pseudo-random numbers by utilizing a logistic map to encrypt HDR images in the format of LogLUV. Chen and Chang [20] prioritized achieving good performance by only encrypting the exponent field bits in OpenEXR images. Chen and Yan [21] presented a scheme for encrypting and authenticating OpenEXR images using torus automorphism and Vernam cipher. Tsai et al. [22] employed random binary digits to encrypt images, and Lan et al. [14] proposed an image encryption algorithm utilizing a 2D sine logistic modulation map (2D-SLMM) and a random permutation technique. In addition, they introduced a set of metrics to comprehensively evaluate the security of the encrypted image in six different aspects.

Research in the combination of data hiding and encryption has also received considerable attention, particularly with the rise of cloud computing and privacy-preserving applications. One such field is reversible data hiding in encrypted images (RDHEI), which can be divided into two approaches: vacating room after encryption (VRAE) and reserving room before encryption (RRBE). In the VRAE approach, Puech et al. [23] used the AES encryption algorithm on 16-pixel image blocks and embedded 1-bit data into a randomly chosen pixel using a pseudo-random number generator. In contrast, for the RRBE approach, Ma et al. [24] reserved room by histogram shifting, significantly increasing the payloads. After two pioneering studies, several improvement schemes have been proposed [25]. For example, Puteaux and Puech [26,27] utilized error prediction and the LOCO-I algorithm in the JPEG-LS compression standard to achieve an average embedding rate of 2.46 bpp. Meanwhile, Wang et al. [28–30] employed a block-level approach for reversible data hiding in the encrypted domain, achieving a maximum embedding rate of 2.5 bpp. These VRAE or RRBE algorithms focus on providing large payloads but fail to evaluate the security of encrypted images.

This paper presents a new adaptive data hiding method which conceals messages when constructing an HDR abstracted image. To the best of our knowledge, few works in the extant literature report on data hiding on abstracted HDR images. Our scheme is suitable to embed secret messages when rendering HDR RGBE and LDR abstracted images. Furthermore, our algorithm performs adaptive data hiding based on the brightness of pixels, where more data are concealed in darker pixels without upsetting the human visual system. Moreover, in order to further protect the image content and the concealed message, we recommend an image encryption algorithm that utilizes a two-dimensional logistic tent modular map (2D-LTMM) [31] and a random permutation scheme [32]. This encryption algorithm is also suitable both for HDR RGBE as well as LDR images. To ensure the security of our image encryption scheme, we evaluated its security using the metrics suggested by Lan et al. [14] and conducted a noise addition analysis.

Our work has the following contributions:

- Our algorithm conceals messages during the image abstraction process which belongs to the constructive image steganography (CIS) approach rather than the SCM approach; it is suitable for both HDR RGBE and LDR images. Our scheme effectively eliminates the vulnerability associated with the SCM approach because the CIS approach offers a significant benefit of producing no cover image. It is difficult for a steganalytic tool to take any advantage to fully train models to distinguish the difference between cover and stego images. Consequently, our scheme can resist training-based steganalysis attack.
- Our adaptive approach conceals a greater number of confidential data in darker and lower luminance pixels, while embedding fewer messages into brighter and higher luminance pixels. This adaptive technique is consistent with the human visual system: dark pixels are less sensitive to the human eye, and minor changes in brighter regions are more apparent. Our algorithm delivers higher embedding rates, ranging from

14% to 33% compared to the current state-of-the-art techniques, while maintaining the quality of a stego image.

- We propose an encryption algorithm that is suitable for both RGBE-based HDR images and LDR images. By utilizing a 2D-LTMM, this scheme produces pseudo-random sequences with improved hyperchaotic characteristics. In addition, exploring a simple yet efficient random permutation scheme enables our scheme to perform not only the bit-level permutation, but also the byte-level diffusion. As a result, our scheme encrypts all bits in all channels in RGBE and LDR images.
- Our encryption algorithm was thoroughly assessed through seven metrics. The outcomes demonstrate that our encryption scheme offers robust security against statistical, differential, brute force, chosen-plaintext, and chosen-key attacks.

The remaining sections of this paper are structured as follows: In Section 2, we elaborate on our proposed methods, which consist of image abstraction, adaptive data embedding, data extraction, image encryption, and image decryption. In Section 3, we outline the experiment results and compare them against other relevant schemes. Lastly, in Section 4, we present our concluding remarks and propose potential avenues for future research.

2. Our Proposed Methods

We describe our scheme in this section. Figure 1 illustrates the flowchart of our algorithm and the output images at each step. Our adaptive data hiding scheme utilizes image abstraction to embed secret messages, thereby constructing an abstracted stego image. The encryption algorithm utilizes a two-dimensional logistic tent modular map and a random permutation scheme to perform bit-level permutation and byte-level diffusion. In Figure 1a, the original input image I undergoes the abstraction process to produce an abstraction result and a quantization result. An image composition process combines edges produced from difference of Gaussian (DoG) filter [9] to produce an intermediate abstracted image, I_A. Moving on to Figure 1b, we utilize I_A as an input for our adaptive data embedding process to construct the abstracted stego image I_S. Next, to ensure the security of the image content and the concealed secret messages, we encrypted I_S using our proposed image encryption algorithm, thereby producing the encrypted image I_E. At the recipient's end, as shown in Figure 1c, stego images are decrypted before extracting secret messages. These processes are detailed as below.

2.1. Image Abstraction

The process of image abstraction involves taking the original image I as input, and transforming it into a new image with a different style while preserving the original content. However, rather than outputting the converting result as an image, we use the intermediate abstracted image I_A as the input for the succeeding steps.

Figure 1a depicts the flowchart of our image abstraction algorithm. The algorithm comprises five distinct steps. The first step is image luminance scaling and leveling (ILSL). This process can appropriately scale the luminance of HDR images for reducing the information of HDR images. After abstracting an image, the luminance of the abstracted image will be reconstructed near the magnitude of the original HDR images. In addition, this process can also be used for LDR images because they have fixed luminance range, which can be considered as a special case of an HDR image.

The second step is image information reduction (IIR). In this phase, we present a new trilateral edge-preserving filter based on the bilateral filter. This new filter can reduce the redundant information within an image and effectively maintain edge information within an image than the existent bilateral and trilateral filters. Through this step, the filtered result can present better gradations between objects. Note that, in an image abstraction algorithm, the effect of the filter will influence the final abstracted results.

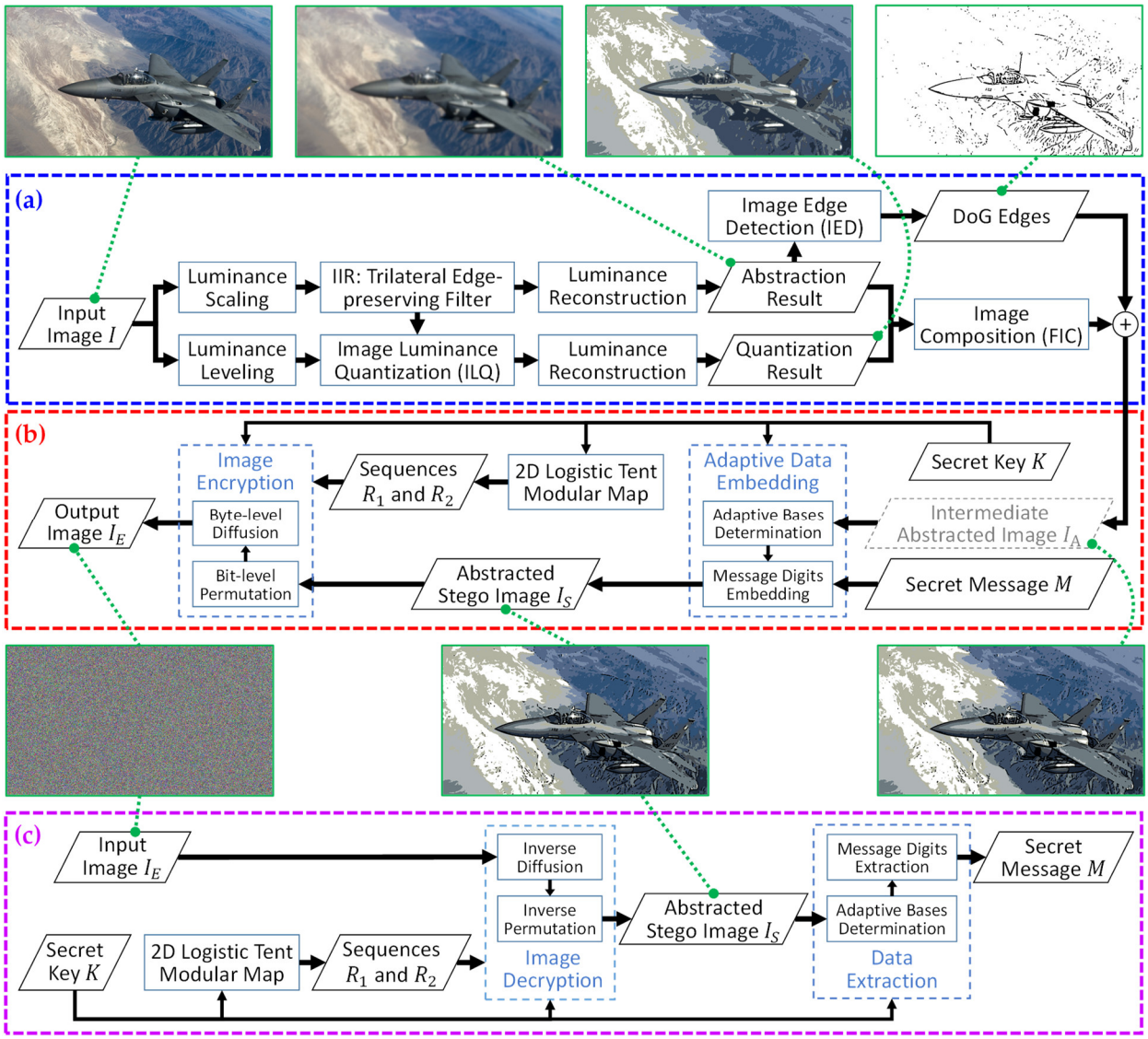


Figure 1. The flowchart of our method. (**a**) The image abstraction process. (**b**) The data embedding and encryption process. Note that the intermediate abstracted image, I_A, is not produced in our scheme due to the constructive data embedding approach; instead, we exhibit it here to clarify the subsequent stages. (**c**) The image decryption and data extraction process.

The third step of image abstraction is image luminance quantization (ILQ). In general, cartoonists have long employed the various color blocks to represent different lighting distributions. To accomplish such an effect, we modify the video abstraction model, encouraging our algorithm to represent the lighting distributions of an image more effectively, making the output image more similar to that of the original image. It also provides an alternative choice for end users to refine their work. Further, differing from [1], our method is capable of automatically determining the suitable luminance level, depending on the luminance distribution appearance in an image.

The next step of image abstraction is image edge detection (IED). Cartoon-based images usually employ some obvious edges to simply exhibit the contour of an object. To simulate this characteristic, we first employ the Difference of Gaussian (DoG) filter to determine the edge information of a target object. We then combine the edge information with the abstracted result, making the abstracted image demonstrate a more visually pleasant appearance.

The final step is fuzzy-relationship-based image composition (FIC). In this step, we balance the influences of both abstracted image and quantized images by computing their fuzzy relationship. The output of these five steps are abstracted images with black contour highlighting the objects ready for data concealment.

2.2. Adaptive Data Embedding

Adaptive data embedding strategies consider local image properties such as texture or luminance. To take advantage of the fact that the human eye is less sensitive to variations in darker pixels, we propose an adaptive message embedding algorithm based on the luminance of pixels for both RGBE and LDR images. The input of this process is the intermediate abstracted image, I_A, and the output is the abstracted stego image I_S, with concealed messages.

Our adaptive data embedding algorithm first determines the adjustable embedding bases. The larger the embedding base, the more messages that can be concealed. We propose a six-level adaptive base approach for HDR images encoded by the RGBE format. For an RGBE pixel $\boldsymbol{P} = (P_R, P_G, P_B, P_E)$, we apply Equation (1) to determine the corresponding embedding base. During the image abstraction process, contour edges are detected and produced, resulting in a small exponent value close to $P_E = 0$ and a scene-referred color channel of approximately 2^{-128}. Regardless of their P_R, P_G, and P_B values, these pixels are indistinguishable to the human eyes. We leverage this fact and assign the maximum base $(b_R, b_G, b_B) = (255, 255, 255)$ to these pixels with $P_E = 0$. The remaining pixels are divided into T groups based on their P_E values, where E_{min} and E_{max} represent the minimum and maximum non-zero P_E values, respectively. The group with the smallest P_E is assigned the embedding base HD_1, while the group with the highest P_E values is assigned embedding base HD_T. In this study, we adopt the parameters of embedding base in descending order associated to their P_E values, where we set $T = 5$ and $(HD_1, HD_2, HD_3, HD_4, HD_5) = (11, 9, 7, 5, 3)$.

$$\begin{aligned} (b_R, b_G, b_B) &= \begin{cases} (255, 255, 255), & \text{if } P_E = 0 \\ (HD_i, HD_i, HD_i), & \text{if } E_{L,i} \le P_E \le E_{R,i}, \text{ where } 1 \le i \le T \end{cases} \\ E_{L,i} &= E_{min} + \tfrac{(i-1)}{T}(E_{max} - E_{min}) \text{ and } E_{R,i} = E_{min} + \tfrac{i}{T}(E_{max} - E_{min}) \end{aligned} \tag{1}$$

For LDR images, we utilize a two-level base approach to segment pixels. Firstly, we calculate the mean values for each channel of the image and denote them as (μ_R, μ_G, μ_B). These values are used as a boundary to conceal different secret messages. They also serve as part of a secret key when extracting the secret message; thus, they are delivering to the receiver from a secure channel. Next, for the i-th pixel, $\boldsymbol{P} = (P_R, P_G, P_B)$, we apply Equation (2) to determine the embedding base (b_R, b_G, b_B) for each channel of the pixel. If the absence of the index i does not result in any ambiguity in the expression, we will not include it. In this study, we adopt the parameters $(LD_1, LD_2, LD_3, LD_4) = (5, 7, 9, 11)$. Nevertheless, end user can change these parameters, depending on the request of the embedding capacity. The larger the parameters, the higher the embedding capacity our scheme can offer:

$$b_R = \begin{cases} LD_1, & \text{for } P_R > \mu_R \\ LD_2, & \text{for } P_R \le \mu_R \end{cases}, \quad b_G = \begin{cases} LD_2, & \text{for } P_G > \mu_G \\ LD_3, & \text{for } P_G \le \mu_G \end{cases}, \quad b_B = \begin{cases} LD_3, & \text{for } P_B > \mu_B \\ LD_4, & \text{for } P_B \le \mu_B \end{cases} \tag{2}$$

Using the determined embedding bases, we decompose the embedding digits from the message stream and inject them into the corresponding channels of the pixels. In the HDR image, a stego pixel is represented as $\boldsymbol{P'} = (P_{R'}, P_{G'}, P_{B'}, P_E)$, where the E channel does not carry any secret message due to its sensitivity to a minor change; while in the LDR image, a stego pixel is represented as $\boldsymbol{P'} = (P_{R'}, P_{G'}, P_{B'})$.

The secret data concealment is based on the modulus operation, as demonstrated in Equation (3), where $k_R, k_G, k_B \in \{-1, 0, 1\}$:

$$\begin{cases} P_{R'} = d_R + P_R - (P_R \ mod \ b_R) + k_R \times b_R, & where \ k_R = arg \ min|P_{R'} - P_R| \\ P_{G'} = d_G + P_G - (P_G \ mod \ b_G) + k_G \times b_G, & where \ k_G = arg \ min|P_{G'} - P_G| \\ P_{B'} = d_B + P_B - (P_B \ mod \ b_B) + k_B \times b_B, & where \ k_B = arg \ min|P_{B'} - P_B| \end{cases} \quad (3)$$

where (d_R, d_G, d_B) represent the hidden secret digits for their corresponding channels. The $arg \ min|\boldsymbol{P'} - \boldsymbol{P}|$ operation ensures that the stego pixel $(P_{R'}, P_{G'}, P_{B'})$ produced by (k_R, k_G, k_B) is as close as possible to the original one, thus reducing the distortion as much as possible. We remark that when embedding message to LDR images, our algorithm sets (μ_R, μ_G, μ_B) as the boundaries, and the pixel values after the message concealment will remain within their original levels without crossing the boundaries to different levels. This means that our algorithm can extract correct messages under the adaptive message embedding strategy.

Taking an LDR pixel as an example for message embedding, we let $\boldsymbol{P} = (P_R, P_G, P_B) = (59, 114, 91)$, $(d_R, d_G, d_B) = (6, 3, 7)$, and $(\mu_R, \mu_G, \mu_B) = (106.91, 100.87, 92.31)$. According to Equation (2), $(b_R, b_G, b_B) = (7, 7, 11)$ represents the determined embedding base. We first use Equation (3) to calculate $P_{R'} = 6 + 59 - (59 \ mod \ 7) + k, k \in \{-7, 0, 7\}$, which results in $P_{R'} = 62$, as it is closer to $P_R = 59$ than the other two candidates, 55 and 69. Similarly, we can derive $P_{G'} = 115$ using the same process. As for $P_{B'} = 7 + 91 - (91 \ mod \ 11) + k, k \in \{-11, 0, 11\}$, which produces three possible values: 84, 95, and 106. Since 95 is the closest one to $P_B = 91$, we might choose it. However, 95 is larger than the level boundary of $\mu_B = 92.31$. Therefore, we choose the value 84, instead, in order to ensure the correctness of message extraction. Finally, the stego pixel value produced is $\boldsymbol{P'} = (P_{R'}, P_{G'}, P_{B'}) = (62, 115, 84)$.

It is worth noting that our adaptive method conveys more secret data in pixels with low E values, which typically appear dark after the tone-mapping process. Similarly, our two-level adaptive LDR image embedding conceals more secret messages in pixels with values below the mean of the corresponding channels. These outcomes are consistent with the human visual sensitivity system that darker pixels are better suited for conveying more messages without attracting human attention, making it less likely for the stego image to be detected by malicious attackers.

2.3. Image Encryption

To secure confidential information and prevent unauthorized access to the image, our scheme involves encrypting the abstracted stego image, I_S, and producing an encrypted version, I_E,. The process has four steps, which are detailed below:

1. Computing the attribute value of the image. We use Equation (4) to obtain the attribute value h of the abstracted stego image:

$$h = 10^a \times \sum_{i=1}^{W \times H} \frac{\left(P_{i,R'} \times 10^{16} + P_{i,G'} \times 10^8 + P_{i,B'}\right)}{10^{24}}, \quad (4)$$

Here, $W \times H$ denotes image resolution; $P_{i,R'}$, $P_{i,G'}$, and $P_{i,B'}$ are values of the channels in pixel i; the parameter a is an integer used to adjust the result h to comply with the restriction, $0.1 < h < 1$. After obtaining the result h, we convert it into a double precision floating-point number and incorporate it into the secret key.

2. The two-dimensional logistic tent modular map (2D-LTMM). We produce two pseudo-random sequences, R_1, R_2, using a 2D-LTMM [31], as shown in Equation (5):

$$\begin{cases} x_{i+1} = \begin{cases} (4ax_i(1-x_i) + 2by_i) \ mod\ 1, & \text{for } y_i < 0.5 \\ (4ax_i(1-x_i) + 2b(1-y_i)) \ mod\ 1, & \text{for } y_i \geq 0.5 \end{cases} \\ y_{i+1} = \begin{cases} (4ay_i(1-y_i) + 2bx_i) mod 1, & \text{for } x_i < 0.5 \\ (4ay_i(1-y_i) + 2b(1-x_i)) \ mod\ 1, & \text{for } x_i \geq 0.5 \end{cases} \end{cases} \quad (5)$$

The constraints of control parameters, a and b, are $a > 0$ and $b > 0$. In this study, we select $(a, b) = (100, 100)$. The trajectory of the 2D-LTMM is shown in Figure 2, which is uniformly distributed over the entire phase plane. In addition, the evaluations of the 2D-LTMM demonstrate a wide and continuous chaotic range, robust chaotic behavior, and hyperchaotic properties [31]. To increase the security of image encryption, the initial values are set as: $(x_0, y_0) = (x_K \times h, y_K \times h)$. The values of (x_K, y_K) are a portion of the secret key. It is worth noting that the initial values (x_0, y_0) are linked to the image's attribute value obtained in Step 1. Therefore, they dynamically change based on the stego image being processed. We remark that this approach provides the benefit of resisting chosen-plain attacks as attackers, who use more plaintext images for encryption, fail to gain any further information to lessen the security of our scheme.

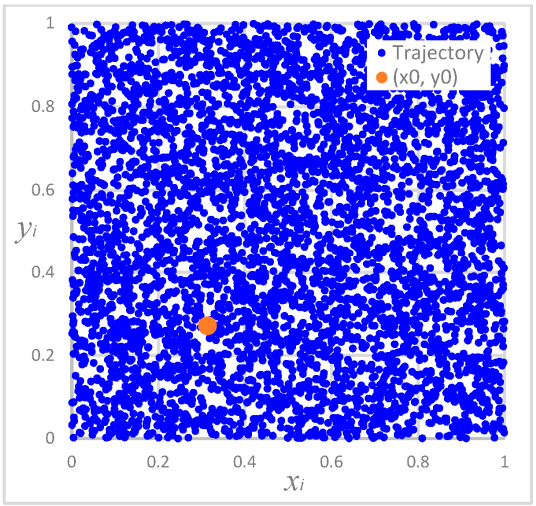


Figure 2. Trajectory of two-dimensional logistic tent modular map using the initial values $(x_0, y_0) = (0.31416, 0.27183)$ and the control parameters $(a, b) = (100, 100)$.

We then apply Equation (5) to generate two pseudo-random sequences: $\boldsymbol{X} = \{x_0, x_1, \ldots, x_q, x_{q+1}, \ldots, x_{q+n_1}\}$ and $\boldsymbol{Y} = \{y_0, y_1, \ldots, y_q, y_{q+1}, \ldots, y_{q+n_2}\}$, where the first q elements in both sequences are discarded to eliminate the initial transient effects. The quantities of (n_1, n_2) are $(32WH, 4WH)$ for HDR images and $(24WH, 3WH)$ for LDR images. The format of each element is the double precision floating point, as defined by the IEEE 754 standard [33]. The least significant 32 bits of each element in $\boldsymbol{X}$ are extracted to form a 32-bit integer sequence, denoted as $\boldsymbol{R_1} = \{R_{1,0}, R_{1,1}, \ldots, R_{1,n_1}\}$. We extract the least significant 8 bits of each element in $\boldsymbol{Y}$ to construct an 8-bit integer sequence, represented as $\boldsymbol{R_2} = \{R_{2,0}, R_{2,1}, \ldots, R_{2,n_2}\}$. The sequence $\boldsymbol{R_1}$ is used in the next step to randomly permute all n_1 bits in the image, and $\boldsymbol{R_2}$ is employed in the following step to diffuse all n_2 bytes of the image.

3. Bit-level permutation for the whole image. This step shuffles the positions of all n_1 bits in the abstracted stego image. To achieve the bit-level permutation, we utilize the random permutation scheme [32], also called the Fisher–Yates Shuffling method, in combination with the sequence $\boldsymbol{R_1}$ generated in Step 2.

Considering an image with n_1 bits, let $\{\mathrm{X}\} = \{1, 2, \ldots, n_1\}$ indicate the indices of bits in the image. During the i-th iteration, we generate an $(n_1 + 1 - i)$-ary random integer by computing $k = R_{1,i}\ mod\ (n_1 + 1 - i)$, where $R_{1,i}$ is the i-th pseudo-random integer in sequence $\boldsymbol{R_1}$. Following this, we swap the $(n_1 + 1 - i)$-th index with the k-th index. The procedure repeats $(n_1 - 1)$ times to rearrange the indices recorded in $\{\mathrm{X}\}$.

We now use an image with only 8 bits ($n_1 = 8$) as an instance. $\{\mathrm{X}\} = \{1, 2, 3, 4, 5, 6, 7, 8\}$ indicates the indices of the bits in the image, and $\{\mathrm{B}\} = \{1, 0, 1, 0, 0, 1, 0, 0\}$ denotes the corresponding bits. With this representation, we can refer to the first three bits as $\mathrm{X}[1] = 1$, $\mathrm{X}[2] = 0$, and $\mathrm{X}[3] = 1$. Table 1 illustrates the permutation process, where the gray color represents the shuffled indices. At the initial round (i = 1), an 8-ary random number $k = 5_8$ is generated from the sequence $\boldsymbol{R_1}$, then the 5th index, 5, is swapped with the 8th index, 8, resulting in shuffled indices $\{1, 2, 3, 4, 8, 6, 7, \mathbf{5}\}$. During the next iteration (i = 2), $k = 3_7$ is generated and the 3rd index, 3, is exchanged with the 7th index, 7, resulting in the indices $\{1, 2, 7, 4, 8, 6, \mathbf{3}, \mathbf{5}\}$. The final permuted indices $\{\mathrm{X}'\}$ are $\{8, 1, 6, 7, 2, 4, 3, 5\}$, which yields a bit-level permutation of $\{\mathrm{B}'\} = \{0, 1, 1, 0, 0, 0, 1, 0\}$.

Table 1. An illustration of random index permutation for an image with 8 bits, where $\{\mathrm{X}\} = \{1, 2, 3, 4, 5, 6, 7, 8\}$ denotes the original order of the bit indices.

i	k	Swap	1st	2nd	3rd	4th	5th	6th	7th	8th
Original index $\{\mathrm{X}\}$			1	2	3	4	5	6	7	8
1	5_8	(5th, 8th)	1	2	3	4	8	6	7	5
2	3_7	(3rd, 7th)	1	2	7	4	8	6	3	5
3	4_6	(4th, 6th)	1	2	7	6	8	4	3	5
4	2_5	(2nd, 5th)	1	8	7	6	2	4	3	5
5	3_4	(3rd, 4th)	1	8	6	7	2	4	3	5
6	0_3	No change	1	8	6	7	2	4	3	5
7	1_2	(1st, 2nd)	8	1	6	7	2	4	3	5
Permuted index $\{\mathrm{X}'\}$			8	1	6	7	2	4	3	5

Our implementation involves first aligning n_1 bits in the image and then applying a bit-level random permutation. It is worth noting that the indices are swapped in place instead of producing a separate copy. As a result, the time complexity is $O(N)$ for the random permutation, and the space complexity is $O(1)$. Furthermore, every element chosen in the array $\{\mathrm{X}\}$ has an equal probability $(1/n_1)$. The output permutation has a probability of $\frac{1}{n_1} \times \frac{1}{n_1 - 1} \times \ldots \times \frac{1}{2} \times \frac{1}{1} = \frac{1}{n_1!}$ when applying this analysis recursively. This implies that every permutation is equally likely to occur.

4. Byte-level diffusion. The last stage involves diffusing the permutated image using the exclusive-OR operator, $\oplus$, in combination with the pseudo-random sequence, $\boldsymbol{R_2}$. The shuffled byte set of the permutated image is denoted as $I_S = \{P_{1,R'}, P_{1,G'}, P_{1,B'}, (P_{1,E'}), \ldots, P_{W\times H,R'}, P_{W\times H,G'}, P_{W\times H,B'}, (P_{W\times H,E'})\}$. Note that an HDR image contains $P_{i,E'}$ elements as it has four channels, including the exponent one. The byte diffusion process comprises of two sub-steps. Firstly, we extract the initial values, $(P_{0,R'}, P_{0,G'}, P_{0,B'}, P_{0,E'}, C_{0,E'})$ for RGBE and $(P_{0,R'}, P_{0,G'}, P_{0,B'}, C_{0,B'})$ for LDR images, from the secret key. Next, we apply Equation (6) to diffuse the bytes of each channel. It is worth noting that as four initial values are provided, the range of index, i, is from 0 to $W \times H - 1$.

$$\begin{cases} C_{i+1,R'} = R_{2,4i} \oplus P_{i+1,R'} \oplus P_{i,R'} \oplus C_{i,E'} \\ C_{i+1,G'} = R_{2,4i+1} \oplus P_{i+1,G'} \oplus P_{i,G'} \oplus C_{i+1,R'} \\ C_{i+1,B'} = R_{2,4i+2} \oplus P_{i+1,B'} \oplus P_{i,B'} \oplus C_{i+1,G'} \\ C_{i+1,E'} = R_{2,4i+3} \oplus P_{i+1,E'} \oplus P_{i,E'} \oplus C_{i+1,B'} \end{cases}, \text{ for RGBE images}$$
$$\begin{cases} C_{i+1,R'} = R_{2,3i} \oplus P_{i+1,R'} \oplus P_{i,R'} \oplus C_{i,B'} \\ C_{i+1,G'} = R_{2,3i+1} \oplus P_{i+1,G'} \oplus P_{i,G'} \oplus C_{i+1,R'} \\ C_{i+1,B'} = R_{2,3i+2} \oplus P_{i+1,B'} \oplus P_{i,B'} \oplus C_{i+1,G'} \end{cases}, \text{ for LDR images} \quad (6)$$

Let us consider a permutated LDR image and take the first pixel as an example. This pixel is represented by the RGB values $\boldsymbol{P}'_1 = \left(P_{1,R'}, P_{1,G'}, P_{1,B'}\right) = (150, 66, 23)$. We use a pseudo-random sequence $\boldsymbol{R_2} = \{10, 153, 117, ...\}$ and four initial values $\left(P_{0,R'}, P_{0,G'}, P_{0,B'}, C_{0,B'}\right) = (79, 51, 114, 99)$ to diffuse the bytes using Equation (6) with $i = 0$. Applying this equation, we get $C_{1,R'} = 10 \oplus 150 \oplus 79 \oplus 99 = 176$. We then use this value to diffuse $C_{1,G'}$ to obtain $C_{1,G'} = 153 \oplus 66 \oplus 51 \oplus 176 = 88$. Similarly, we diffuse $C_{1,B'} = 72$ using $C_{1,G'}$. Finally, the stego pixel $\boldsymbol{P}'_1$ is diffused to become $\boldsymbol{C}'_{\boldsymbol{1}} = (176, 88, 72)$.

One advantage of our scheme is the avalanche effect [34,35]. This effect means that if an error occurs in one pixel during encryption, it will trigger a chain reaction that spreads the error to all subsequent pixels. The error spreading causes a significant change in the encrypted image, making it difficult to analyze using differential attack methods.

2.4. Image Decryption

We need to decrypt the image before revealing the hidden messages. This involves taking the encrypted image, I_E, as input and producing the deciphered stego image, I_S, as output. The decryption process is detailed as follows:

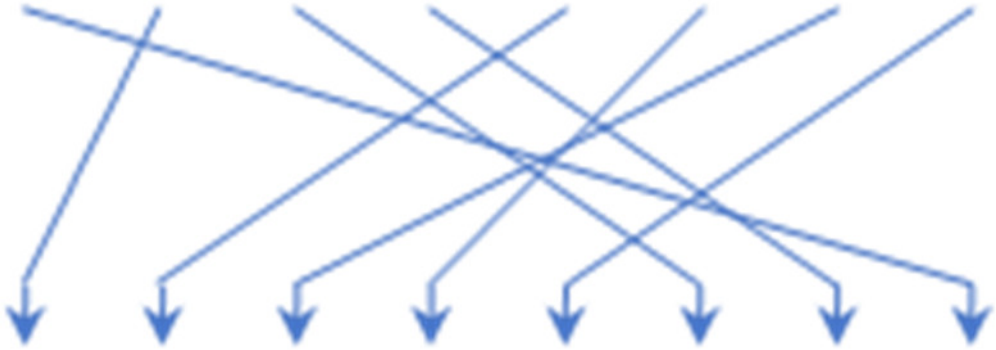
1. Regenerating the pseudo-random sequences using the 2D logistic tent modular map. We retrieve the values of (h, q, x_K, y_K) from the secret key and follow Step 2 of the image encryption to regenerate the same pseudo-random sequences $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$.
2. Byte-level inverse diffusion. Applying $\boldsymbol{R_2}$ with the initial values, $\left(P_{0,R'}, P_{0,G'}, P_{0,B'}, P_{0,E'}, C_{0,E'}\right)$ for RGBE and $\left(P_{0,R'}, P_{0,G'}, P_{0,B'}, C_{0,B'}\right)$ for LDR images, we can utilize Equation (7) to decipher stego encrypted values in all channels:

$$\begin{cases} P_{i+1,R'} = R_{2,4i} \oplus C_{i+1,R'} \oplus P_{i,R'} \oplus C_{i,E'} \\ P_{i+1,G'} = R_{2,4i+1} \oplus C_{i+1,G'} \oplus P_{i,G'} \oplus C_{i+1,R'} \\ P_{i+1,B'} = R_{2,4i+2} \oplus C_{i+1,B'} \oplus P_{i,B'} \oplus C_{i+1,G'} \\ P_{i+1,E'} = R_{2,4i+3} \oplus C_{i+1,E'} \oplus P_{i,E'} \oplus C_{i+1,B'} \end{cases}, \text{for RGBE images}$$
$$\begin{cases} P_{i+1,R'} = R_{2,3i} \oplus C_{i+1,R'} \oplus P_{i,R'} \oplus C_{i,B'} \\ P_{i+1,G'} = R_{2,3i+1} \oplus C_{i+1,G'} \oplus P_{i,G'} \oplus C_{i+1,R'} \\ P_{i+1,B'} = R_{2,3i+2} \oplus C_{i+1,B'} \oplus P_{i,B'} \oplus C_{i+1,G'} \end{cases}, \text{for LDR images} \quad (7)$$

As a continuation of our previous example, assume that we have generated a pseudo-random sequence $\boldsymbol{R_2} = \{10, 153, 117, \ldots\}$, and the four initial values $\left(P_{0,R'}, P_{0,G'}, P_{0,B'}, C_{0,E'}\right) = (79, 51, 114, 99)$. Suppose that the first encrypted stego pixel is $\boldsymbol{C}'_{\boldsymbol{1}} = (176, 88, 72)$. To decrypt this pixel, we use Equation (7) and derive $P_{1,R'} = 10 \oplus 176 \oplus 79 \oplus 99 = 150$ for the red component. By applying the same approach, we can decrypt the other two components and obtain the deciphered stego pixel $\boldsymbol{P}'_1 = (150, 66, 23)$.

3. Bit-level inverse permutation. This step applies the sequence $\boldsymbol{R_1}$ generated in Step 1 and the random permutation scheme to inversely restore bits in an image. After this step, the bit order will return to the original one. Following up on the previous 8-bit image example, we can obtain the indices of permutated bits $\{\mathrm{X}'\} = \{8, 1, 6, 7, 2, 4, 3, 5\}$, representing the permuted bits $\{\mathrm{B}'\} = \{0, 1, 1, 0, 0, 0, 1, 0\}$. Table 2 illustrates the operation in this step. The first bit in $\{\mathrm{B}'\}$ is 0, and its corresponding index is 8. Thus, we place this bit in the 8th position of the output buffer. Similarly, the second bit, 1, is placed in the 1st position of the output buffer, etc. Finally, we produce the inversely permutated bits $\{\mathrm{B}\} = \{1, 0, 1, 0, 0, 1, 0, 0\}$, which has the same order before the permutation process.

Table 2. An illustration of inverse permutation in bit-level, where $\{X'\} = \{8,1,6,7,2,4,3,5\}$ represent the permutated indices, and $\{B'\} = \{0,1,1,0,0,0,1,0\}$ are the permutated bits.

Position	1st	2nd	3rd	4th	5th	6th	7th	8th
Permutated bits $\{B'\}$	0	1	1	0	0	0	1	0
	↓	↓	↓	↓	↓	↓	↓	↓
Permutated indices $\{X'\}$	8	1	6	7	2	4	3	5
								
Inversely permutated bits $\{B\}$	1	0	1	0	0	1	0	0

2.5. Data Extraction

Once the stego image has been decrypted, the embedded messages can be extracted. The first step of message extraction is to determine the embedding base.

When dealing with HDR images in the RGBE format, we employ the six-level adaptive base technique, shown in Equation (1), which uses the intact E values to determine the embedding bases for the R, G, and B channels of the pixel. In the case of LDR images, we utilize the values (μ_R, μ_G, μ_B) provided in the secret key and apply Equation (2) to determine the embedding base (b_R, b_G, b_B) for each channel of the pixel.

Once the embedding bases have been determined, the secret digits (d_R, d_G, d_B) embedded in the R, G, and B channels of a stego pixel can be extracted using Equation (8). These secret digits are later used to produce the original binary secret bits:

$$(d_R, d_G, d_B) = (P_{R'} \bmod b_R, P_{G'} \bmod b_G, P_{B'} \bmod b_B) \tag{8}$$

Continuing from our previous example in data embedding, the decrypted stego pixel $\boldsymbol{P'} = (P_{R'}, P_{G'}, P_{B'}) = (62, 115, 84)$, and the corresponding mean values (μ_R, μ_G, μ_B) produced by the secret key are $(106.91,\ 100.87,\ 92.31)$. First, we apply Equation (2) to determine the embedding base $(b_R, b_G, b_B) = (7, 7, 11)$, and then we use Equation (8) to extract the secret digits: $(d_R, d_G, d_B) = (62\ mod\ 7, 115\ mod\ 7, 84\ mod\ 11) = (6, 3, 7)$. In this way, our scheme correctly extracts secret messages that have been concealed in the stego pixel, $\boldsymbol{P'}$.

3. Experimental Results and Analysis

We implemented our algorithm in the C++ programming language and conducted experiments on a notebook computer equipped with a Ryzen 7 3750H CPU, 16 GB memory, and the operating system is Windows 10. To evaluate our algorithm, we used five HDR RGBE images (HDR 1 to HDR 5) which had been used as test images in HDR data embedding literature [11–15]. In addition, we adopted five standard LDR color images (LDR 1 to LDR 5) collected from the USC-SIPI image database which have been utilized in several data hiding or steganographic schemes [16,17,36]. Our original test images are presented in Figure 3. Additionally, all HDR images presented in this paper were tone mapped

using [37] prior to display for visualization purpose. In addition, we use tone-mapped images for image quality assessments presented later in this section.

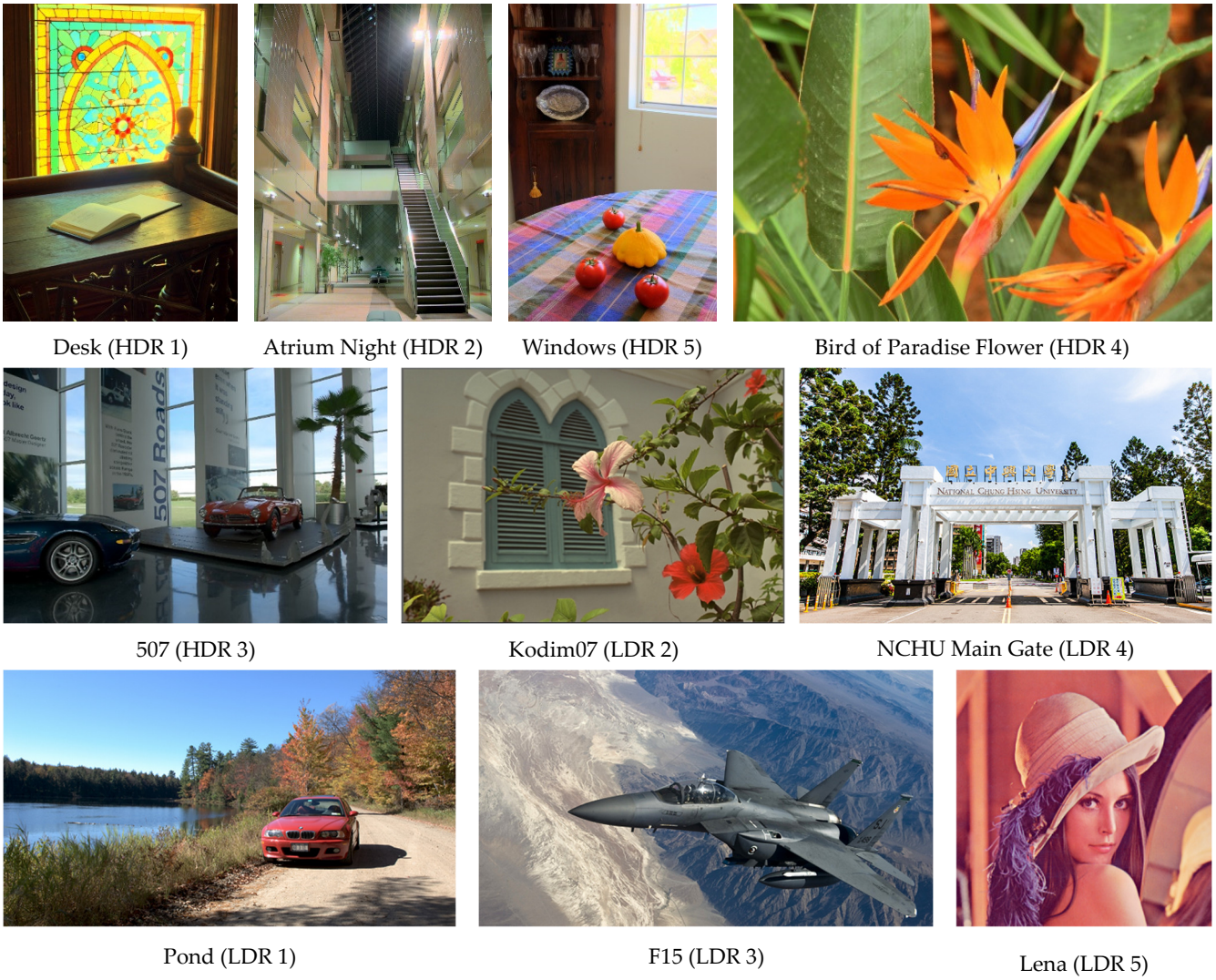
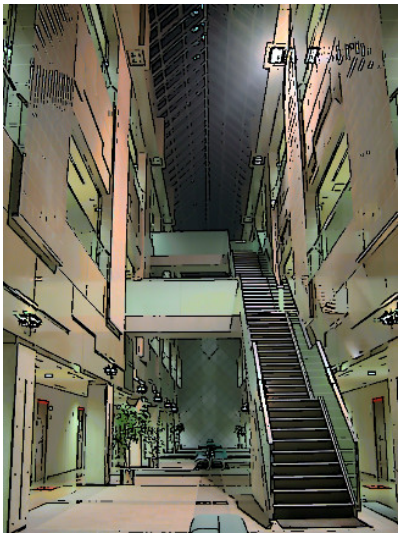


Figure 3. The presentation of the original test images.

3.1. Embedding Results of Secret Messages

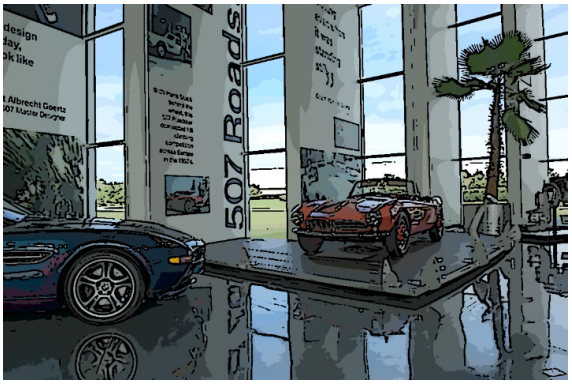
Figure 4 and Table 3 show the abstracted embedded outcomes of our test images, comprising five RGBE and five LDR images. In Table 3, the embedding capacity (EC) varies between 2.30 million bits (for LDR 5) to 33.50 million bits (HDR 4), while the embedding rate (ER) varies from 8.332 bpp (HDR 4) to 12.428 bpp (HDR 1). The PSNR values between the tone-mapped original abstracted HDR image and the tone-mapped stego HDR images are over 43 dB. In addition, the PSNR values between the original abstracted LDR image and the stego LDR images are all greater than 37 dB. These high PSNR values exhibit good image quality. Additionally, the structural similarity index measure (SSIM) values, which were derived on the similar definition for HDR and LDR images, are very close to 1.0 for the two types of test images. The statistics demonstrate that our scheme can convey a large number of secret messages while producing stego images with excellent image quality.

Desk (HDR 1)



Atrium Night (HDR 2)

Windows (HDR 5)



507 (HDR 3)

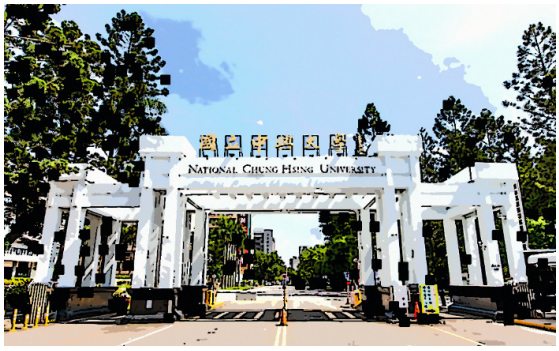
Bird of Paradise Flower (HDR 4)

Pond (LDR 1)

Kodim07 (LDR 2)

Lena (LDR 5)

F15 (LDR 3)



NCHU Main Gate (LDR 4)

Figure 4. The exhibition of abstracted stego images.

Table 3. Test images and the payloads.

ID	Name	Resolution	EC (bits)	ER (bpp)	* PSNR (dB)	* SSIM
HDR 1	Desk	644 × 874	6,995,448	12.428486	45.506	0.999875
HDR 2	Atrium Night	760 × 1016	8,523,403	11.038390	45.399	0.999836
HDR 3	507	1072 × 712	8,487,159	11.119559	50.949	0.999736
HDR 4	Bird of Paradise Flower	2464 × 1632	33,503,279	8.331563	47.327	0.999845
HDR 5	Windows	524 × 800	4,658,477	11.112780	43.934	0.999314
LDR 1	Pond	1072 × 603	5,731,397	8.866421	39.213	0.995143
LDR 2	Kodim07	768 × 512	3,461,004	8.801788	38.960	0.996886
LDR 3	F15	960 × 540	4,597,217	8.868089	39.533	0.995428
LDR 4	NCHU Main Gate	1024 × 634	5,660,622	8.719166	37.400	0.969324
LDR 5	Lena	512 × 512	2,295,636	8.757156	39.087	0.996439

* For HDR images, the values are tone-mapped results.

It is worth noting that HDR images have a higher embedding rate and better image quality than LDR ones. This is because the abstracted HDR images have a great number of black edge pixels, where their exponent channel is $P_E = 0$. Our algorithm takes advantage of this phenomena to embed nearly 24 bpp using our proposed six-level adaptive embedding scheme. Additionally, the distortion caused by data embedding is significantly reduced after the tone-mapping operation. As a result, the embedding rate and quality assessment results for HDR images are better than those for LDR images, which only use a two-level adaptive embedding scheme without conducting any tone-mapping operation. However, the test image HDR 4 is an exception which offers a smaller embedding rate, because it has significantly fewer number of black edge pixels. Furthermore, the pixel distribution skews towards higher E-channel values. Consequently, the overall embedding rate for the test image, HDR 4, is smaller than that of other test HDR images.

3.2. Analysis of Security

This section presents a performance evaluation of our image encryption algorithm, which is designed to encrypt the stego image, I_S, and generate an encrypted image, I_E. We adopt the security metrics proposed by Lan et al. [14], which include visual perception, histogram analysis, correlation analysis, entropy, image sensitivity, and key security. Additionally, we conduct noise addition analysis to evaluate the security of our proposed scheme.

3.2.1. Perception by Vision

Figure 5 displays the images produced by our algorithm at various stages. The HDR RGBE case is shown first followed by the case for the LDR image. Figure 5a,b show the original HDR image by directly display it without tone-mapped operation and the tone-mapped results, respectively. Figure 5c exhibits the abstracted stego images concealed around 8.487 million bits of hidden messages, while Figure 5d shows the contours of the abstracted stego images. Figure 5e shows the encrypted tone-mapped image which successfully shelters the outlines and detail features without revealing any meaningful information. Finally, Figure 5f shows the image decryption result. Figure 5g–j are the corresponding results for the LDR case. The visual perception demonstrates that our encryption scheme is visually secure.

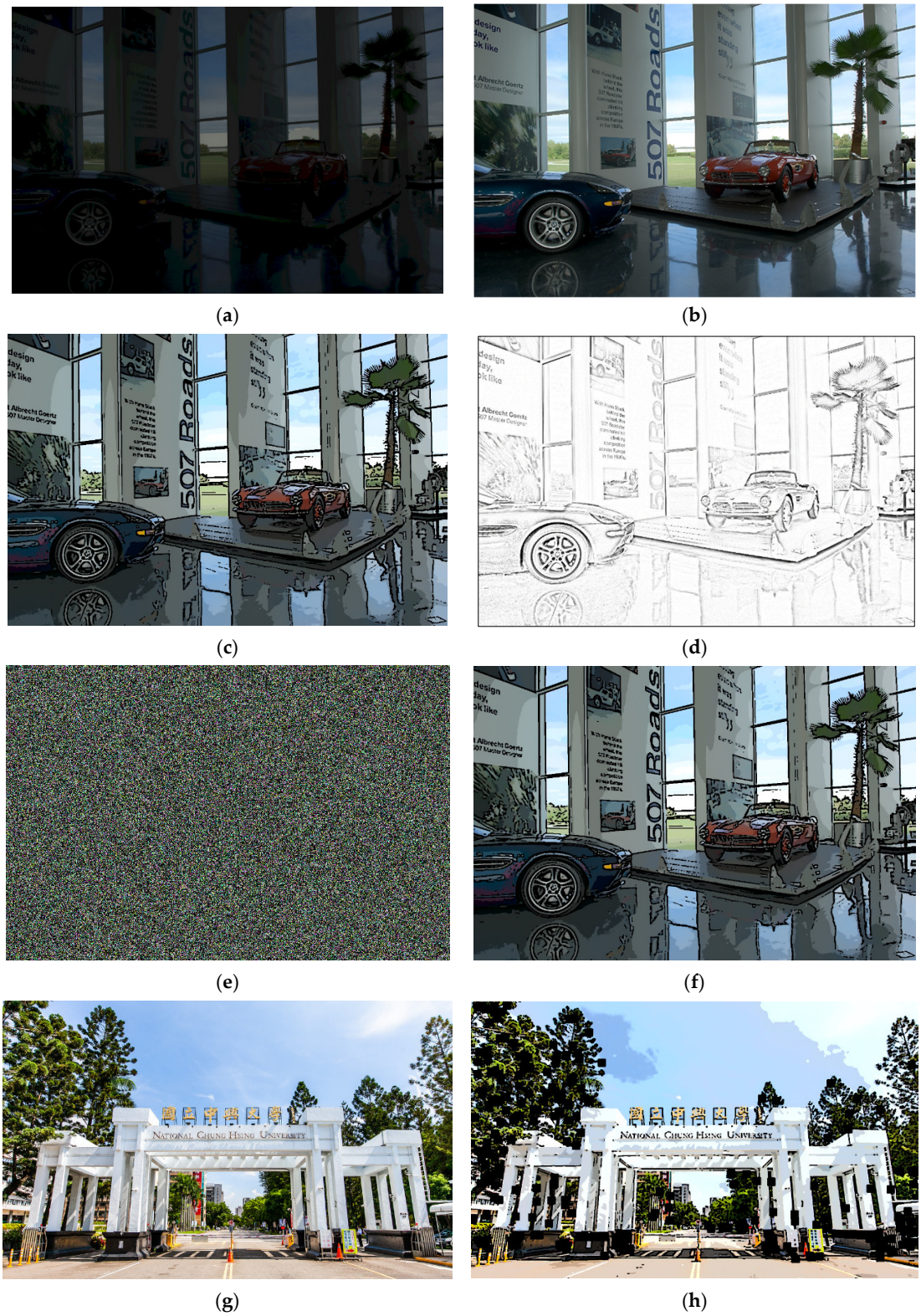
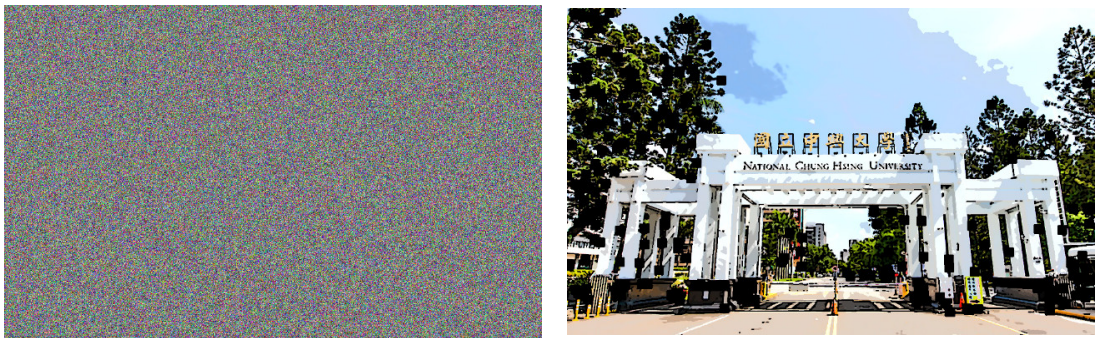


Figure 5. *Cont.*

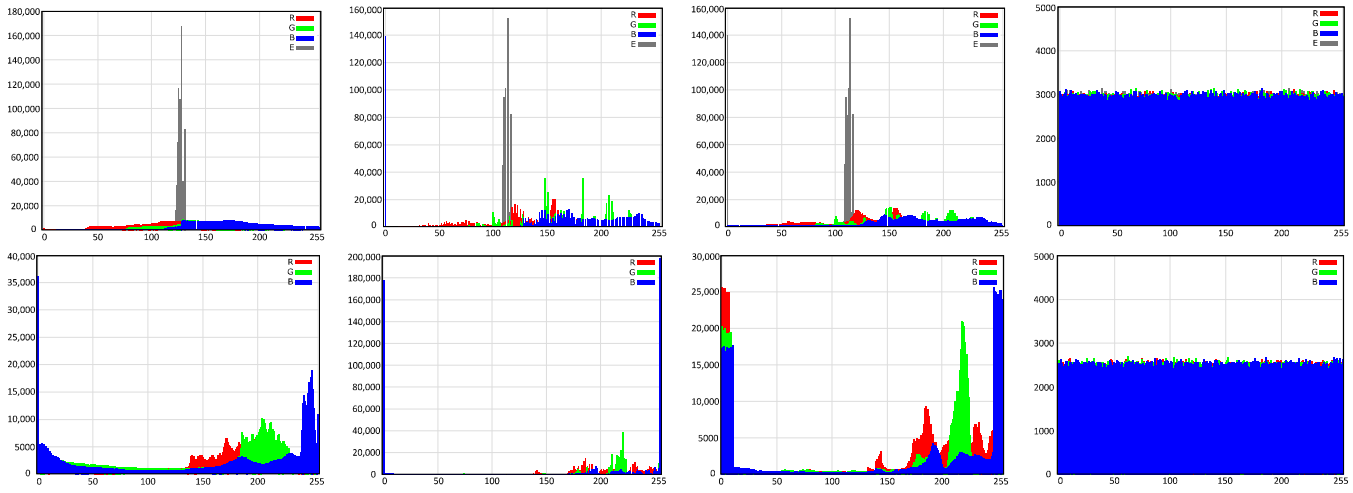


(**i**) (**j**)

Figure 5. The visual perception of the HDR test image "507" (**a**–**f**) and LDR test image "NCHU Main Gate" (**g**–**j**). (**a**) Directly displaying the original HDR image without any tone-mapping (TM) operation. (**b**) The tone-mapped HDR image. (**c**) The abstracted stego image concealed with around 8.487 million secret bits. (**d**) The edges of the abstracted stego image. (**e**) The ciphered image. (**f**) The deciphered TM image. (**g**–**j**) The corresponding results of LDR test image "NCHU Main Gate," where the abstracted stego image (**h**) has concealed more than 5.66 million secret bits.

3.2.2. Analysis of Histogram

An image's histogram provides valuable statistical information, making it a prime target for statistical attacks by malicious eavesdroppers. A well-designed encryption algorithm should hide statistical information and prevent them from being revealed. Figure 6 displays histograms of the original, intermediate abstracted, abstracted stego, and ciphered images. Our encryption algorithm makes the histogram of abstracted stego uniform and completely different from that of the plaintext in all the channels, making it impossible to reveal any meaningful information. The results of the histogram analysis indicate that the proposed method is effective in preventing statistical attacks.



(**a**) (**b**) (**c**) (**d**)

Figure 6. Histogram analysis of the encryption scheme on images, "507," (top row) and the "NCHU Main Gate," (bottom row). (**a**) The histograms of the original images. (**b**) The histograms of the intermediate abstracted images. (**c**) The histograms of the abstracted stego images. (**d**) The histograms of the encrypted stego images.

The uniformity of a histogram can be quantitatively evaluated using the Variance of Histogram (VOH), which can be computed using Equation (9). The equation uses

$Z = \{z_0, z_1, \ldots, z_{255}\}$ to represent the number of counts in each bin, while μ is the mean count of all bins. A smaller VOH indicates a more uniform histogram.

$$VOH(Z) = \sum_{i=0}^{255} E\left[(z_i - \mu)^2\right] \tag{9}$$

We use the Chi-square (χ^2) test, in addition to VOH, to determine any statistically significant difference between the expected and observed frequencies of the histogram. If considering the present scenario, the degree of freedom is 255. Assuming a significance level (α) of 0.05, the threshold would be 293.25. If the calculated χ^2 value from the histogram is less than the given threshold, it would lead us to accept the null hypothesis, which indicates that the histogram of the stego abstracted image is statistically indistinguishable from a uniform distribution.

In Table 4, we have presented the values of VOH and χ^2 for two sample stego abstracted images, namely "507" and "NCHU Main Gate". The table presents these values in both plain and ciphered status, offering a comparison of VOH and χ^2 values before and after encryption. In particular, VOH values drop to less than 4×10^{-3}% of the plaintext, and χ^2 values are less than the threshold value (293.25) in all the channels. Moreover, we have performed χ^2 test for all 10 test images to gain more insights, and the results are plotted in Figure 7. From the graph, it is evident that the χ^2 values of all 10 test images in every channel are below the dashed-line threshold.

Table 4. Variance of Histogram (VOH) and χ^2 values for the abstracted stego images.

Image	Channel	VOH		χ^2 Value	
		Plain	Ciphered	Plain	Ciphered
507 (HDR)	R	78,703,629.406	2810.023	6,757,715.622	224.635
	G	79,139,306.367	2668.469	6,795,124.075	289.866
	B	80,909,521.609	3241.375	6,947,119.749	249.191
	E	281,099,137.109	3114.953	24,135,964.816	292.041
NCHU Main Gate (LDR)	R	121,376,328.734	2373.852	12,252,500.062	239.632
	G	124,007,220.664	2610.609	12,518,079.058	263.532
	B	133,176,246.500	2740.508	13,443,658.953	276.644

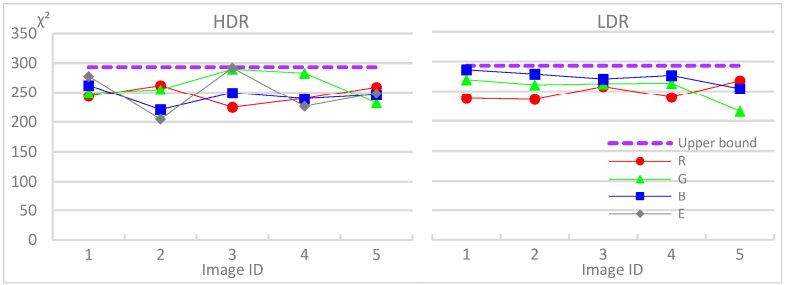


Figure 7. All 10 abstracted-encrypted stego images pass the χ^2 test in all channels.

Based on the findings displayed in Table 4 and Figure 7, we can conclude that our algorithm significantly reduces VOH and χ^2 values during the encryption process. In addition, the output images generated by our encryption algorithm passed the statistical hypothesis test. Our scheme ensures that the histogram of the encrypted image in all channels has a uniform distribution, thus providing a high level of security for the ciphered images.

3.2.3. Analysis of Correlation

Adjacent pixels in plain images are typically highly correlated, making it feasible to predict a pixel using the nearby pixels. To prevent malicious attacks, a suitable encryption

algorithm should eliminate as many as possible those correlations. We can measure the correlation among pixels using the correlation coefficient ($r_{x,y}$) specified in Equation (10):

$$r_{x,y} = \frac{E(x - E(x))E(y - E(y))}{\sqrt{D(x)}\sqrt{D(y)}} \tag{10}$$

In this equation, x and y represent the adjacent pixel pair, while $E(x)$ and $D(y)$ correspond to the expectation and variance of x and y over these samples, respectively. The value of $r_{x,y}$ ranges from -1 to +1, with values closer to zero indicating lesser correlation between paired pixels. Moreover, when $r_{x,y} > 0$, it indicates a direct correlation between the pixels, whereas $r_{x,y} < 0$, the pixels are inversely related.

We calculate the $r_{x,y}$ for R, G, and B channels horizontal (H), vertical (V), and diagonal (D) directions, using 5000 pairs of adjacent pixels. For an RGBE image, we compute the $r_{x,y}$, which are all floating-point values in the R, G, and B channels. The results for the plain and ciphered images are presented in Table 5. The statistics indicate that the $r_{x,y}$ have been substantially reduced in the encrypted image and its values are approaching zero. This outcome demonstrates the success of our encryption algorithm in disrupting the pre-existing correlation in the plain image.

Table 5. Correlation coefficients of the adjacent pixels in the abstracted stego test images.

Image	Channel	Plain Image			Ciphered Image		
		H	V	D	H	V	D
507 (HDR)	R	0.974378	0.969364	0.950080	−0.000235	−0.000080	0.000381
	G	0.976883	0.974866	0.956821	0.000765	−0.000685	0.000299
	B	0.979750	0.980885	0.964439	−0.000883	0.000147	0.000373
NCHU Main Gate (LDR)	R	0.908312	0.905556	0.855767	−0.000568	0.000030	0.001296
	G	0.915412	0.913782	0.868304	−0.001012	0.001337	0.000797
	B	0.937499	0.937616	0.903753	−0.000181	0.000799	−0.000592

Figure 8 displays the correlation coefficients of all 10 encrypted abstracted stego test images. The results are found to be very close to zero, indicating that the proposed encryption algorithm has significantly reduced the correlation in all three directions, thereby improving the security of the encrypted images.

3.2.4. Analysis of Entropy

Information entropy is a concept in information theory that quantifies the amount of uncertainty or randomness in a set of data or a signal. It is also known as the Shannon entropy [38]. In the case of an 8-bit grayscale image, denoted by Z, the gray level i is represented by z_i. Equation (11) defines the Shannon entropy, where $P(z_i)$ represents the probability of gray level i occurring:

$$H(Z) = -\sum_{i=0}^{255} P(z_i) log_2[P(z_i)] \tag{11}$$

Assuming that Z represents a perfect random image, where $P(z_i) = 1/256$ and it has the entropy of 8. Thus, a well-encrypted 8-bit image will have an entropy close to 8. Table 6 presents the entropy analysis results of our encryption algorithm. The plain image exhibits low entropy value, particularly in the E-channel, with a minimum value of 3.16 in the HDR image. However, the statistics of the ciphered image indicate that the entropy values in all the channels are close to 8. The results indicate that our scheme is effective in producing a ciphered image with entropy values that closely resemble those of an ideal random image.

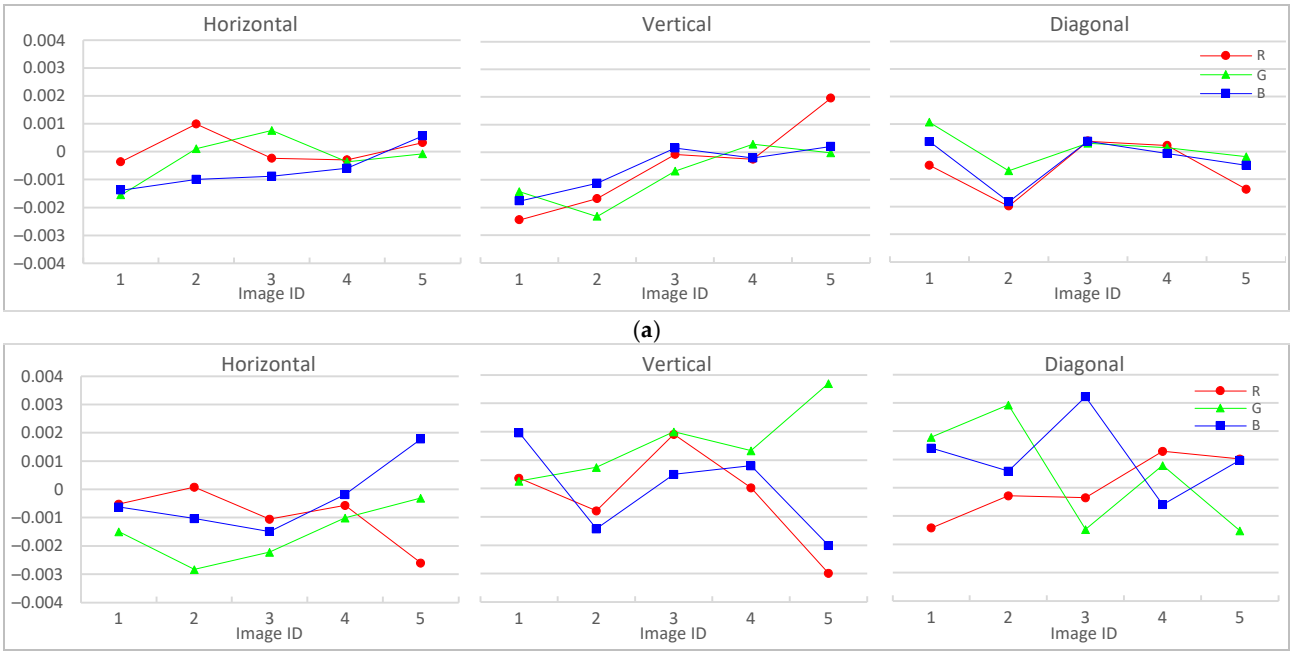


Figure 8. Correlation coefficients between adjacent pixels are close to zeros for all 10 encrypted abstracted stego test images in three channels and three directions. (**a**) HDR images; (**b**) LDR images.

Table 6. Analysis of entropy for the abstracted stego test images in the plain and ciphered statuses.

Image	Channel	Plain Image	Ciphered Image
507 (HDR)	R	7.368588	7.999787
	G	7.329590	7.999727
	B	7.216471	7.999764
	E	3.161784	7.999724
NCHU Main Gate (LDR)	R	6.182650	7.999734
	G	5.950375	7.999707
	B	5.882892	7.999693

Moreover, Wu et al. [39] introduced the local Shannon entropy (LSE) to address the limitations of the Shannon entropy, aka the global one. The LSE for an 8-bit grayscale image Z, can be calculated by averaging the information entropy over k non-overlapping and randomly chosen n-pixels blocks, B_i. The computation of LSE is shown in Equation (12). If the parameters $(k, n) = (30, 1936)$ and significance level (α) is 0.05, a ciphered image is expected to have an LSE value close to the ideal of 7.9025. To satisfy the test criteria, the LSE value should be within the interval [7.9019, 7.9030]

$$\overline{H_{k,n}}(Z) = \sum_{i=1}^{k} \frac{H(B_i)}{k} \tag{12}$$

In order to evaluate our algorithm, we calculated the average LSE values across 10 encrypted stego images, as depicted in Figure 9. The results indicate that all LSE values fall within the target range, thus satisfying the statistical test. The tests conducted on both global and local entropy indicate that our encryption scheme generates ciphered images that display characteristics of randomness on both global and local levels, thereby making them resilient against statistical attacks.

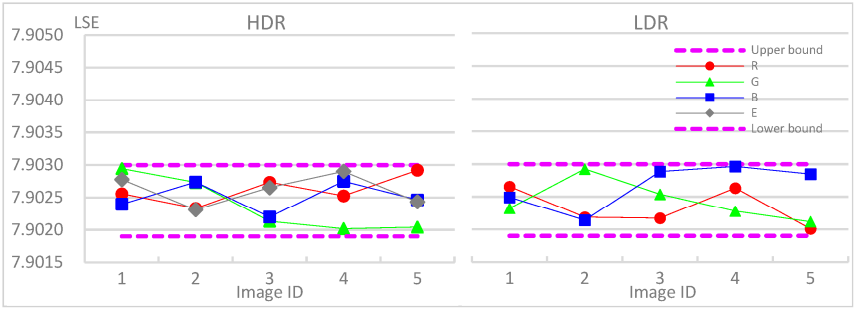


Figure 9. The test results of the local Shannon entropy for all 10 ciphered abstracted stego images.

3.2.5. Image Sensitivity Analysis

Differential attacks are a crucial process in undermining the encryption security because a malicious hacker is likely to reveal vital information from two ciphered images containing a subtle change in their plaintext images. A competent encryption algorithm must be able to detect the slightest variation between two input images, even a single bit of difference. The sensitivity of an image is an indicator of its ability to thwart differential attacks. The most commonly used metrics to evaluate image sensitivity are the number of pixel change rates (NPCR) and the unified average changed intensity (UACI) [40–42]. Equations (13)–(15) define NPCR and UACI. These equations utilize the pixel values of input images I_1 and I_2, denoted as $I_1(i,j)$ and $I_2(i,j)$:

$$D(i,j) = \begin{cases} 0 \; if \; I_1(i,j) = I_2(i,j) \\ 1 \; if \; I_1(i,j) \neq I_2(i,j) \end{cases} \tag{13}$$

$$NPCR(I_1, I_2) = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \tag{14}$$

$$UACI(I_1, I_2) = \frac{1}{W \times H} \left(\sum_{i,j} \frac{|I_1(i,j) - I_2(i,j)|}{255} \right) \times 100\% \tag{15}$$

The NPCR and UACI results are reported in Table 7. All statistics are close to the ideal value of 99.6094 and 33.4635 [41], respectively, indicating that our encryption scheme can effectively defeat the differential attack.

Table 7. NPCR and UACI statistics for two encrypted abstracted stego images.

Image	Channel	NPCR (%)	UACI (%)
507 (HDR)	R	99.602208	33.456771
	G	99.607187	33.461968
	B	99.608628	33.462320
	E	99.605300	33.471379
NCHU Main Gate (LDR)	R	99.604538	33.470771
	G	99.604538	33.468483
	B	99.606941	33.447217

We note that the values of NPCR and UACI are influenced by the resolutions of the input images. Moreover, the optimal range of NPCR/UACI values that would render the ciphered image immune to malicious attacks remains uncertain. To address this issue, Wu et al. [41] established the criteria for statistical hypothesis tests by calculating the means and standard deviations of NPCR and UACI.

To perform the hypothesis test, we used 10 abstracted stego test images. For each image, we altered the least significant bit (LSB) of the R-channel in the center and the four corner pixels, generating five corresponding stego images that differed by one bit from the original. After image encryption, we calculated the NPCR and UACI values for each image

pair, the original and the altered one, to analyze their statistical significance. Figure 10 shows the NPCR and UACI results, averaged over five test cases.

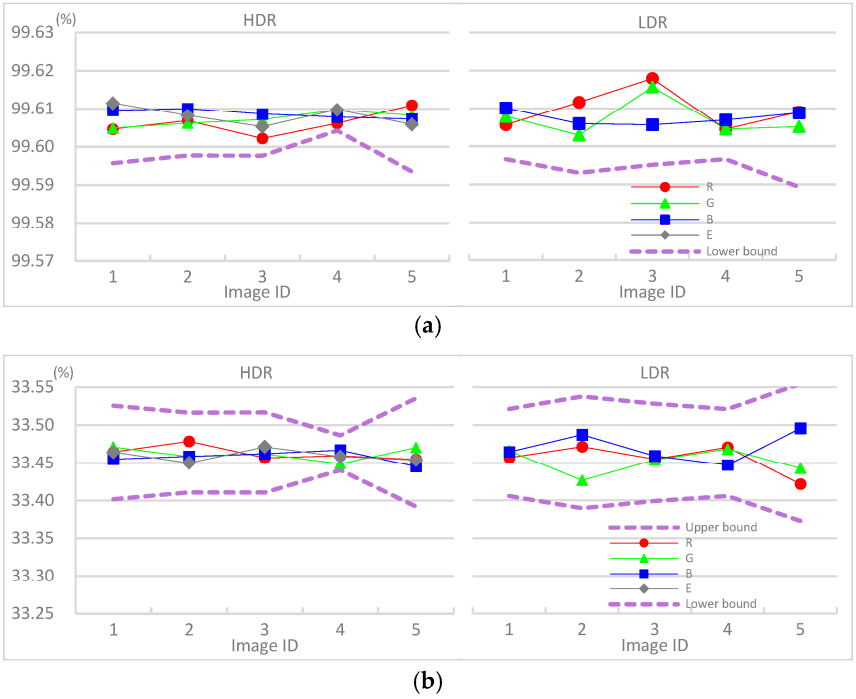


Figure 10. The statistical tests results of (**a**) NPCR and (**b**) UACI for all encrypted abstracted stego test images.

Results shown in Table 7 and Figure 10 indicate that our encryption scheme exhibits a significant degree of image sensitivity, making it capable of withstanding differential attacks. This conclusion is supported by the fact that all NPCR values are greater than the threshold, and all UACI results fall within the bounds defined by the lower and upper limits, indicating that our scheme has successfully passed both the NPCR and UACI statistical tests.

3.2.6. Key Security

A suitable encryption algorithm must have a large key space to prevent brute force attacks. Additionally, it should be responsive to even subtle changes in the key, resulting in a completely different ciphered image in order to resist chosen key attacks. Key space and key sensitivity are two perspectives from which the security of a key can be analyzed:

- **Key space:** When applying a 2D-LTMM to produce two pseudo-random sequences, three floating-point numbers, each consisting of 64 bits and in double-precision format, are utilized: h, x_K, and y_K. Moreover, in order to eliminate the initial transient effect, we discard the first q items of the sequences, where q is represented by a 16-bit integer. In addition, during the encryption process, we adopt 8-bit integers as the initial values: $\left(P_{0,R'}, P_{0,G'}, P_{0,B'}, P_{0,E'}, C_{0,E'}\right)$ for HDR RGBE and $\left(P_{0,R'}, P_{0,G'}, P_{0,B'}, C_{0,B'}\right)$ for LDR images. Therefore, the key space exceeds the minimum requirement of 2^{128}, as it is greater than 2^{240}. The analysis shows that our scheme requires large key space; this provides the ability to resist brute force attacks.
- **Key sensitivity:** The sensitivity of the key in our algorithm was evaluated as follows: Initially, two keys, K_1 and K_1, were generated with a single-bit difference. We then applied K_1 and K_2 to encrypt the same abstracted stego image I_S, thus producing I_{E1} and I_{E2}, which are shown in Figure 11b,d, respectively. Next, using K_1, we decrypted I_{E1} and were able to obtain the original image, as demonstrated in Figure 11e. However, if we attempt to decrypt I_{E1} using the other key, K_2, the output appears similar to a

noisy image, as shown in Figure 11f. Analogous outcomes are attained in Figure 11g,h if we decrypt I_{E2} using K_2 and K_1, respectively. Furthermore, we analyzed NPCR and UACI between I_{E1} and I_{E2} in Figure 11c. The results are in close proximity to the ideal values of 99.6094 and 33.4635, indicating that our algorithm remains sensitive to secret keys, even when only a single bit differs between them. Consequently, the key sensitivity analysis confirms that our scheme can withstand chosen key attacks.

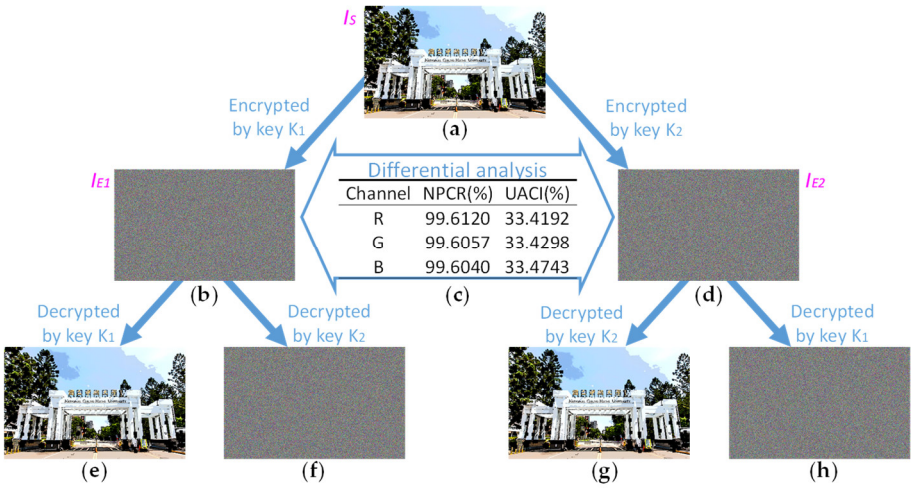


Figure 11. The test of key sensitivity using the image "NCHU Main Gate," with secret key K_1 differing from secret key K_2 by only one bit. (**a**) The original abstracted stego image I_S. (**b**) I_S ciphered with K_1, producing I_{E1}. (**c**) NPCR and UACI values between the image pair (I_{E1}, I_{E2}). (**d**) I_S ciphered with K_2, producing I_{E2}. (**e**) I_{E1} decrypted by K_1 to obtain the original I_S. (**f**) Decryption of I_{E1} using the incorrect key, K_2, produces a noisy image. (**g**) Decryption of I_{E2} by K_2 successfully recovers I_S. (**h**) Decryption of I_{E2} using the incorrect key, K_1, also produces a noisy image.

3.2.7. Robustness to Noise Addition Analysis

In this section, we evaluate the robustness of our proposed method under different types of noise added to stego images. Specifically, we apply 1% and 5% of bit flip (BF) to HDR images. For LDR images, we add 1% and 5% salt-and-pepper (S&P) noise to each channel.

Table 8 presents the bit error rates (BER) obtained from four test images when extracting secret bits from stego images with added noise. A smaller BER indicates less impact of added noise on the secret bits. The results indicate that more noise added leads to a higher BER. Additionally, there is more impact of noise when the stego image has a higher embedding rate (ER). This is because larger embedding bases are more susceptible to noise. For instance, HDR 2 has an ER of 12.43 bpp, higher than HDR 4 (8.33 bpp), results in a higher BER.

Table 8. The bit error rate and image quality under the noise addition attacks.

Image	Type	BER (%)	PSNR	SSIM
HDR 2	BF (1%)	0.9287	32.6236	0.9879
	BF (5%)	4.5601	25.2547	0.9485
HDR 4	BF (1%)	0.9049	31.0547	0.9920
	BF (5%)	4.4381	24.8969	0.9696
LDR 1	S&P (1%)	0.8868	24.6057	0.8680
	S&P (5%)	4.3696	17.6689	0.6353
LDR 2	S&P (1%)	0.8912	24.9598	0.8800
	S&P (5%)	4.3763	18.1616	0.6752

In addition to the bit error rates, Table 8 presents the image quality results. The reference images used are the stego images without any attacks. The PSNR values increase as the noise ratios decrease. The experimental results indicate that HDR images have better image quality than LDR images. This is because the impact of noise addition is significantly reduced after the tone-mapping process, thereby producing better image quality.

3.3. A Comparison with the Latest Related Works

In Table 9, we compare our algorithm with other approaches that currently represent the forefront of research in adaptive data hiding for HDR images. Our scheme is the only algorithm that can support the image abstraction. In addition, while [11–13,15] concealed secret messages by modifying the existing cover images, our scheme and that in [14] adopted constructive data hiding, where secret messages are embedded by constructing a stego image directly. Our scheme and that in [14] can support image encryption to further protect the hidden messages.

Table 9. An evaluation of adaptive data hiding in comparison with state-of-the-art methods.

Scheme	Proposed	[14]	[13]	[15]	[12]	[11]
Year	2023	2022	2020	2017	2011	2009
Input Image	RGBE/LDR	OpenEXR	RGBE	OpenEXR	RGBE	RGBE
Image Abstraction	**Yes**	No	No	No	No	No
Constructive	**Yes**	**Yes**	No	No	No	No
Encryption	**Yes**	**Yes**	No	No	No	No
ER (bpp)	8.332–12.428	7.30–9.29	0.490–2.292	2.433–20.002	0.1256–0.1281	5.04–9.70
PSNR (dB)	37.400–50.949	32.60–51.90	50.65–51.77	45.12–82.32	N.A.	30.00–40.00
SSIM	0.9693–0.9999	0.994–1.000	0.854–0.995	0.7572–0.9999	N.A.	N.A.

In terms of the embedding rate, it ranges from 0.5 to 20.0 bpp, with most competitors being outperformed by our algorithm, except for that in [15] since the encoding bit length in an HDR RGBE is 32, while that in HDR OpenEXR is 48. In addition, the space to conceal secret message in RGBE is far less than that in OpenEXR, since the Exponential channel is required to be intact for message concealment in order to avoid significant pixel distortion.

Our algorithm generates a moderate level of image quality, where a tone-mapped stego image has the PSNR of 50.949 dB, lower than [15] but close to [13,14]. It is worth noting the PSNR is influenced by tone-mapping algorithms as well as the number of hidden secret messages. Our algorithm offers a larger embedding rate than most competitive schemes, while still producing the stego image with good quality exhibiting moderate PSNR values and high SSIM values. We conclude that our approach surpasses the current state-of-the-art methods.

Table 10, presents a comparison between our proposed encryption scheme and state-of-the-art algorithms for HDR images. Upon closer examination, these schemes did not undergo a thorough security analysis. In contrast, our proposed scheme has conducted a comprehensive security evaluation process, consisting of seven different metrics. Undoubtedly, security is the most crucial aspect of an encryption algorithm. The lack of comprehensive security analysis leaves room for doubt that the ciphered images can effectively withstand malicious attacks, such as differential and statistical attacks [34,35].

Table 10. A comparison of security metrics on HDR image encryption schemes.

Scheme	Proposed	[14]	[22]	[21]	[20]	[19]	[18]
Year	2023	2022	2022	2021	2014	2014	2013
Input Image	RGBE/LDR	OpenEXR	RGBE	OpenEXR	OpenEXR	LogLuv	RGBE
Visual Perception	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Histogram	**Yes**	**Yes**	No	**Yes**	**Yes**	No	**Yes**
Correlation	**Yes**	**Yes**	No	**Yes**	**Yes**	No	No
Entropy	**Yes**	**Yes**	No	No	No	No	No
Image Sensitivity	**Yes**	**Yes**	No	No	No	No	No
Key Security	**Yes**	**Yes**	No	No	No	No	**Yes**
Noise Addition	**Yes**	No	No	No	No	No	No

4. Conclusions and Future Work

We presented a constructive adjustable image data hiding scheme for HDR RGBE images and LDR images, where secret messages are concealed by directly synthesizing a stego abstracted image. Depending on the E channel in an HDR image, our six-level adaptive message embedding approach can conceal various secret messages. A two-level adaptive technique was presented for LDR images, where adaptivity is on the basis of the mean value of each channel. This encourages our scheme to convey more data in lower luminance pixels and fewer in higher luminance ones without upsetting the human visual system. Without degrading the quality of the stego image, our scheme achieves an embedding rate that is 14% to 33% higher than current state-of-the-art schemes.

A novel image encryption method was introduced to protect the hidden messages carried in an abstracted image. Our new scheme incorporates a 2D logistic tent modular map with a bit-level random permutation technique. The former has a continuous chaotic range, robust chaotic behavior and hyperchaotic properties, while the latter is flexible and exhibits the time complexity and the space complexity of $O(N)$ and $O(1)$, respectively. The proposed algorithm supports excellent security and is robust against statistical, differential, brute force, and chosen key attacks. Our proposed adaptive data hiding and secure encryption scheme is feasible in seeking to extend the HDR and LDR data hiding applications.

Possible future work includes extending our algorithm to provide the feature of reversibility, which includes the ability to recover the initial image followed by message extraction.

Author Contributions: Conceptualization, C.-F.L. and C.-M.W.; methodology, C.-F.L.; software, C.-F.L.; validation, C.-F.L.; writing—original draft preparation, C.-F.L. and C.-M.W.; writing—review and editing, C.-F.L. and C.-M.W.; project administration, C.-M.W. and W.L.; Supervision, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available in a publicly accessible repository.

Acknowledgments: This work is supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 109-2221-E-005-062, MOST 110-2221-E-005-069, and MOST 111-2221-E-005-076.

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

References

1. Winnemöller, H.; Olsen, S.C.; Gooch, B. Real-time video abstraction. *ACM Trans. Graph.* **2006**, *25*, 1221–1226. [CrossRef]
2. Ma, Y.; Basu, A.; Li, X. RAST: Restorable arbitrary style transfer via multi-restoration. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–7 January 2023; pp. 331–340. [CrossRef]
3. 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, VT, USA, 2010.
4. Kim, M.; Kautz, J. Consistent tone reproduction. In Proceedings of the 10th IASTED International Conference on Computer Graphics and Imaging, (CGIM 2008), Innsbruck, Austria, 13–15 February 2008; pp. 152–159.

5. Ward, G.J. The RADIANCE lighting simulation and rendering system. In Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, Orlando, FL, USA, 24–29 July 1994; pp. 459–472. [CrossRef]
6. Larson, G.W. LogLuv encoding for full-gamut, high-dynamic range images. *J. Graph. Tools* **1998**, *3*, 15–31. [CrossRef]
7. OpenEXR. Available online: https://www.openexr.com (accessed on 1 April 2023).
8. Wang, L.; Yoon, K.-J. Deep learning for HDR imaging: State-of-the-art and future trends. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *40*, 8874–8895. [CrossRef]
9. Zhou, Y.-C.; Huang, M.-L.; Wang, C.-M. High dynamic range image stylization. *Int. J. Adv. Inf. Technol.* **2007**, *1*, 21–39.
10. Kumar, P.; Poornima, B.; Nagendraswamy, S.; Manjunath, C.; Rangaswamy, B. HDR and image abstraction framework for dirt free line drawing to convey the shapes from blatant range images. *Multidim. Syst. Signal Process.* **2022**, *33*, 401–458. [CrossRef]
11. Cheng, Y.-M.; Wang, C.-M. A novel approach to steganography in high-dynamic-range images. *IEEE Multimed.* **2009**, *16*, 70–80. [CrossRef]
12. Yu, C.-M.; Wu, K.-C.; Wang, C.-M. A distortion-free data hiding scheme for high dynamic range images. *Displays* **2011**, *32*, 225–236. [CrossRef]
13. Gao, X.; Pan, Z.; Gao, E.; Fan, G. Reversible data hiding for high dynamic range images using two-dimensional prediction-error histogram of the second time prediction. *Signal Process.* **2020**, *173*, 107579. [CrossRef]
14. Lan, C.-F.; Wang, C.-M.; Lin, W. XtoE: A novel constructive and camouflaged adaptive data hiding and image encryption scheme for high dynamic range images. *Appl. Sci.* **2022**, *12*, 12856. [CrossRef]
15. Lin, Y.-T.; Wang, C.-M.; Chen, W.-S.; Lin, F.-P.; Lin, W. A novel data hiding algorithm for high dynamic range images. *IEEE Trans. Multimed.* **2017**, *19*, 196–211. [CrossRef]
16. Hseih, K.-S.; Wang, C.-M. Constructive image steganography using example-based weighted color transfer. *J. Inf. Secur. Appl.* **2022**, *65*, 103126. [CrossRef]
17. Hsieh, K.-S.; Wang, C.-M. Multi-hider reversible data hiding using a weighted color transfer and modulus operation. *Appl. Sci.* **2023**, *13*, 1013. [CrossRef]
18. Yan, J.-Y.; Chen, T.-H.; Lin, C.-H. Encryption in high dynamic range images for RGBE format. In Proceedings of the Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Beijing, China, 16–18 October 2013; pp. 493–496. [CrossRef]
19. Lin, K.-S.; Chen, T.-H.; Lin, C.-H.; Chang, S.-S. A tailor-made encryption scheme for high-dynamic range images. *Adv. Intell. Syst. Comput.* **2014**, *238*, 183–192. [CrossRef]
20. Chen, T.-H.; Chang, S.-S. Image encryption on HDR images for OpenEXR Format. *Int. J. Eng. Sci.* **2014**, *4*, 19–23.
21. Chen, T.-H.; Yan, J.-Y. Commutative encryption and authentication for OpenEXR high dynamic range images. *Multimed. Tools Appl.* **2021**, *80*, 27807–27828. [CrossRef]
22. Tsai, Y.-Y.; Liu, H.-L.; Kuo, P.-L.; Chan, C.-S. Extending multi-MSB prediction and Huffman coding for reversible data hiding in encrypted HDR Images. *IEEE Access* **2022**, *10*, 49347–49358. [CrossRef]
23. Puech, W.; Chaumont, M.; Strauss, O. A reversible data hiding method for encrypted images. In Proceedings of the SPIE, San Jose, CA, USA, 18 March 2008; Volume 6819. [CrossRef]
24. Ma, K.; Zhang, W.; Zhao, X.; Yu, N.; Li, F. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 553–562. [CrossRef]
25. Puteaux, P.; Ong, S.-Y.; Wong, K.-S.; Puech, W. A survey of reversible data hiding in encrypted images—The first 12 years. *J. Vis. Commun. Image Represent.* **2021**, *77*, 103085. [CrossRef]
26. Puteaux, P.; Puech, W. An Efficient MSB Prediction-Based Method for High-Capacity Reversible Data Hiding in Encrypted Images. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1670–1681. [CrossRef]
27. Puteaux, P.; Puech, W. A Recursive Reversible Data Hiding in Encrypted Images Method with a Very High Payload. *IEEE Trans. Multimed.* **2021**, *23*, 636–650. [CrossRef]
28. Wang, X.; Chang, C.-C.; Lin, C.-C. Reversible data hiding in encrypted images with block-based adaptive MSB encoding. *Inf. Sci.* **2021**, *567*, 375–394. [CrossRef]
29. Wang, X.; Chang, C.-C.; Lin, C.-C.; Chang, C.-C. Privacy-preserving reversible data hiding based on quad-tree block encoding and integer wavelet transform. *J. Vis. Commun. Image Represent.* **2021**, *79*, 103203. [CrossRef]
30. Wang, X.; Chang, C.-C.; Lin, C.-C.; Chang, C.-C. Reversal of pixel rotation: A reversible data hiding system towards cybersecurity in encrypted images. *J. Vis. Commun. Image Represent.* **2022**, *82*, 103421. [CrossRef]
31. Hua, Z.; Zhu, Z.; Yi, S.; Zhang, Z.; Huang, H. Cross-plane colour image encryption using a two-dimensional logistic tent modular map. *Inf. Sci.* **2021**, *546*, 1063–1083. [CrossRef]
32. Durstenfeld, R. Algorithm 235: Random permutation. *Commun. ACM* **1964**, *7*, 420. [CrossRef]
33. *IEEE STD 754-2019*; IEEE Standard for Floating-Point Arithmetic. IEEE Computer Society: Piscataway, NJ, USA, 2019; pp. 1–84. [CrossRef]
34. Singh, K.N.; Singh, A.K. Towards integrating image encryption with compression: A survey. *ACM Trans. Multimedia Comput. Commun. Appl.* **2022**, *18*, 89. [CrossRef]
35. Ghadirli, H.M.; Nodehi, A.; Enayatifar, R. An overview of encryption algorithms in color images. *Signal Process.* **2019**, *164*, 163–185. [CrossRef]

36. Setiadi, E.R.I.M.; Rustad, S.; Andono, P.N.; Shidik, G.F. Digital image steganography survey and investigation (goal, assessment, method, development, and dataset). *Signal Process.* **2023**, *206*, 108908. [CrossRef]
37. Reinhard, E.; Stark, M.; Shirley, P.; Ferwerda, J. Photographic tone reproduction for digital images. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive (SIGGRAPH '02), New York, NY, USA, 1 July 2002; pp. 267–276. [CrossRef]
38. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Technol. J.* **1948**, *27*, 379–423. [CrossRef]
39. Wu, Y.; Zhou, Y.; Saveriades, G.; Again, S.; Noonan, J.P.; Natarajan, P. Local Shannon entropy measure with statistical tests for image randomness. *Inf. Sci.* **2013**, *222*, 323–342. [CrossRef]
40. Chen, G.; Mao, Y.; Chui, C. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals* **2004**, *21*, 749–761. [CrossRef]
41. Wu, Y.; Noonan, J.P.; Agaian, S. NPCR and UACI randomness tests for image encryption. *Cyber J. Multidisci. J. Sci. Technol. J. Select. Areas Telecommun.* **2011**, *2*, 31–38.
42. Zhang, Y. Statistical test criteria for sensitivity indexes of image cryptosystems. *Inf. Sci.* **2021**, *550*, 313–328. [CrossRef]