Secure high capacity tetris-based scheme for data hiding

Guo-Dong Su1,2, Chin-Chen Chang3, Chia-Chen Lin3,4 5, Zhi-Qiang Yao6
1Engineering Research Center for ICH Digitalization and Multi-source Information Fusion (Fujian Polytechnic of Fujian Normal University), Fujian Province University, Fuzhou 350500, People’s Republic of China
2Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
3Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan
4Department of Computer Science and Information Management, Providence University, Taichung 40724, Taiwan
5College of Mathematics and Informatics, Fujian Normal University, Fuzhou Fujian 350117, People’s Republic of China
6E-mail: ally.cclin@ncut.edu.tw

Abstract: Information hiding is a technique that conceals private information in a trustable carrier, making it imperceptible to unauthorised people. This technique has been used extensively for secure transmissions of multimedia, such as videos, animations, and images. This study proposes a novel Tetris-based data hiding scheme to flexibly hide more secret messages while ensuring message security. First, an $L_Q \times L_Q$ square lattice $Q$ is selected to determine the maximum embedding capacity, and then it is filled without gaps through rotating and sliding tetrominoes while making the shape of each tetromino different. Secondly, according to the decided $Q$, the reference matrix and corresponding look-up table are constructed and then used for secret messages embedding and extraction. In the authors approach, each pixel pair of the original image can be processed to conceal 4- or 6-bit secret messages. The experimental results show that their proposed Tetris-based scheme has excellent performance, exceeding the performance of some state-of-the-art schemes in both embedding capacity and visual quality. The proposed scheme also provides secure covert communication.

1 Introduction

Data hiding (DH) is one technique that has been designed to hide secret messages using various cover media, i.e. images, videos, and audio files, for secret and secure communication [1-4] or the error resilience of multimedia communications [5-7]. The key point of DH lies on the imperceptible modifications made on cover media so that the human eyes have difficulty detecting any distortion caused by DH.

In the past decade, many studies in terms of DH have been presented and have achieved quite good visual quality and a promising high embedding capacity (EC) of stego-images. Many literature works have indicated that there still exists a trade-off caused by DH.

In 2006, an improved LSB (least significant bit) substitution scheme [8], namely the LSB matching revisited scheme, was proposed by Mielikainen [9]. His scheme concealed 2-bit secret messages into a pixel pair, where the LSB of the first pixel carries one bit and a function of those two-pixel values carries another bit. In the same year, Zhang and Wang [10] observed that there was room for improvement in fully exploiting the directions of modification in Mielikainen’s scheme [9]. Thus, they proposed the EMD (exploiting modification direction) scheme, where $n$ cover pixels as a unit can carry a $(2n+1)$-ary secret digit, and only one pixel is added or subtracted by 1 at most. EMD's advantage is that it successfully controlled the distortion while its hiding capacity still has not been enhanced. In 2010, the schemes of Kim et al. [11], also called EMD-2 and 2-EMD, were proposed to promote the EC. Experimental results confirmed that Kim et al.’s embedding rate (ER) is higher than the original EMD scheme while maintaining a similar stego-image quality. Meanwhile, by extension, i.e. EMD-k and k-EMD, their scheme is also easy to be conducted. The following year, Kieu and Chang [12] proposed a new DH scheme by exploiting eight modification directions to hide several secret messages into a pixel pair at a time. In this way, it provides a high EC up to 4.5 bpp (bits per pixel), and the corresponding image quality is around 31.70 dB.

Recently, a series of DH schemes [13-19] based on the concept of a special structure, i.e. turtle shell (TS), were also developed to obtain a better balance between EC and stego-image quality. To enhance the level in terms of security, some researches [20-29] have explored using the idea of a puzzle game to hide messages. The review of them [13-29] will be presented in Section 2. However, those schemes might still have difficulty achieving a win-win in both EC and image quality. Now, this scenario presents a requirement for the continuous innovations of the DH technique.

In this paper, another Tetris-based DH scheme is proposed to further enhance both the EC and image quality of puzzle game-based DH scheme. Tetris is a tetromino-based falling a puzzle game whose main point is to fill a gameboard by rotating and sliding seven tetrominoes. In our proposed Tetris-based scheme, first, a square lattice $Q$ with the size of $L_Q \times L_Q$ is filled without gaps through rotating and sliding these tetrominoes. Secondly, a reference matrix sized 256 × 256 is generated by replicating this decided $Q$ and then the corresponding look-up table (LUT) is constructed. Afterwards, according to this reference matrix and LUT, each pixel pair can be processed to embed flexible secret messages. The larger the $L_Q$, the higher the EC has. The experimental results show that the proposed scheme achieves a considerable EC along with satisfactory image quality than that of other puzzle game-based DH schemes.

The rest of this paper is organised as follows. Section 2 gives the related works. Details of the proposed Tetris-based DH scheme are discussed in Section 3. Experimental results are presented in Section 4, and Section 5 presents our conclusions.

2 Related works

To provide sufficient background knowledge, several TS, Sudoku and Tetris-based DH schemes will be briefly reviewed in Section 2.1. Next, we introduce a TS-based scheme [14] in Section 2.2
2.1 Review of the related DH schemes

In 2014, Chang et al. [13] proposed a novel DH scheme based on the concept of a special structure, i.e. TS. In their scheme, an 8-ary secret digit was concealed into a pixel pair of cover image with the guidance of TS. Their scheme provides a good visual quality, i.e. 49.40 dB, of stego-image on average along with a high ER of 1.5 bpp. To further enhance the EC for Chang et al.’s TS-based scheme, a DH scheme combing the reference matrix and a location table was presented by Liu et al. [14] in 2016. In Liu et al.’s scheme, each pixel pair conceals 4-bit secret messages, and the average image quality of the stego-images is about 45.55 dB. In 2017, Jin et al. [15] used a particle swarm optimisation algorithm to reduce the distortion from the original image. Compared to schemes [13, 14], the average improvement of peak signal-to-noise ratios (PSNRs) of Jin et al.’s scheme was 0.03 and 0.01 dB, respectively. At the same year, Liu et al. [16] extended the structure of TS-based reference matrix into various models to meet different ECs and the needs of image quality. With the guidance of an extended reference matrix, an N-ary secret digit can be embedded into a pixel pair. Their scheme provides a high EC with the ER of 2.5 bpp and good image quality with an average PSNR of 41.87 dB. Afterwards, in order to overcome the problem of the drastic decrease in stego-image quality as N increases in [16], in 2019, Liu et al. [17] further proposed a DH scheme based on a multi-matrix structure of TS to balance the requirements of EC and image quality. For different smooth-component image blocks, the different TS-based reference matrices were employed to implement the DH to ensure as few distortions as possible. Their scheme achieves an ER of 2.35 bbb, while the image quality is around 44.31 dB. At the same year, Li et al. [18] defined an upgraded reference matrix based on TS by adding 8 to partial digits to ensure digits of the three neighbouring TsSs ranged from 0 to 15. Meanwhile, with the help of the self-defined numbering rule and LUT, each pixel pair can carry 6-bit secret messages, while maintaining an average PSNR of 40.93 dB. Also, in 2020, a DH scheme based on a 3D magic cube was developed by Lee et al. [19], where they achieved an ER of 2.25 bpp and average PSNR of 44 dB.

In addition, some research studies [20–29] have explored using the idea of a puzzle game to hide messages. In 2008, Chang et al. [20] proposed a DH scheme based on a number placement puzzle game, i.e. Sudoku, in which a pixel pair of the original image was utilised to carry a 9-ary secret digit with the help of a 256 × 256 reference matrix that can be constructed by a determined solution of Sudoku. The average ER provided by scheme in [20] is about 1.58 bpp, and the stego-image visual quality is around 44.90 dB. In the same year, Hong et al. [21] observed that there was space for improvement in searching for candidate elements in Chang et al.’s scheme [20]. That is, in some cases, a couple more suitable candidate pixel pairs in the reference matrix could be found for DH, with fewer distortions. As a result, scheme [21] achieves better image quality in the stego-image than that of Chang et al.’s scheme [20], with an average improvement of 2.60 dB. Later, Hong et al. [22] again provided another suggestion, which was to measure the difference between the original pixel pair and the candidate pixel pair by using the Euclidean distance instead of the Manhattan distance. In this scenario, the stego-image quality is 0.60 dB higher than their previous work proposed in [21]. Also, Chang et al. [23] presented an improved version of the ones present in [20–22] using the greedy method, where PSNR reaches 48.14 dB. In 2009, Farn and Chen [24] proposed a new DH scheme using jigsaw puzzle images. In their scheme, the image was first divided into non-overlapping blocks. For each block, a semicircle was drawn and attached to the right and bottom sides. Then, the secret messages were embedded through adjusting the attached positions and the orientations of semicircles. The EC of a 1024 x 1024 image offered by their scheme is 992 bytes. In 2014, a steganographic method based on the Tetris game on a practical scenario was proposed by Ou and Chen [25]. The secret messages were carried by using a generated tetromino sequence. Their scheme is undetectable and has a weakness in EC. In 2018, Lyu et al. [26] presented a DH scheme based on kind of irregularly shaped Sudoku. A benefit of the reshaped Sudoku is that their scheme provides more secure than that of the original Sudoku-based scheme. Furthermore, their scheme also achieves a PSNR of 47.48 dB when the ER is set to 1.5 bpp. To increase the EC of Sudoku-based scheme, the single-layer and multi-layer mini-Sudoku matrix-based high capacity schemes were individually presented by He et al. [27] and Chen et al. [28]. Those two schemes inherit the idea of coordinate mapping used in scheme [14]. Benefit by this, the former scheme achieved the ER of 2.0 bpp and the PSNR of 46.37 dB on hiding capacity and image quality, respectively, and the latter one reached the ER of 3.0 bpp and the PSNR of 40.01 dB on hiding capacity and image quality, respectively. Additionally, Hong et al. [29] developed the multidimensional mini-Sudoku-based DH scheme in 2020. It achieves good image quality and EC, which are like the results in [27].

2.2 Liu et al.’s TS-based DH scheme

In 2016, a high capacity TS-based DH scheme was proposed by Liu et al. [14], where each pixel pair is processed to conceal a 4-bit secret message. In their scheme, a reference matrix RM is constructed in advance, as shown in Fig. 1, and then shows the renewal strategy during data embedding. To reduce the caused distortion and minify the search area, RM must possess the following features: (i) The RM has an x-axis (p_x) and a y-axis (p_y) +1. Each of them ranges from 0 to 255 and maps to the pixel value of a given pixel pair in a cover image. (ii) The difference of two adjacent pixel values in the same row of RM always remains 1, but the differences between two adjacent pixel values in the same column are 2 and 3 in turn. (iii) RM consists of a series of interconnected TSs, and each TS covers eight different distinct digits that all range in [0, 7]. Meanwhile, the digits located inside the TS are named as back elements, such as the red digits shown in Fig. 1; conversely, other digits located on the edge of the TS are called edge elements, as the blue digits shown in Fig. 1.

In Liu et al.’s scheme, there are 16 different cases in total that have been determined according to four pre-defined locations and different digits to enhance the EC. Based on those cases, a location table LT as shown in Fig. 2 is constructed and used for DH. For simplicity, each element in LT is called as an LT-shape, and its corresponding digit d represents the value of LT-shape. To carry secret messages (s_j, s_k), a pixel pair is changed to a renewed pixel pair with the principle of minimum Euclidean distance. The renewed pixel pair should satisfy two properties: (i) its projected location in RM should be the same as the LT-shape LT(s_j, s_k); (ii)
Taking a specific case as an example, assume that we want the pixel pair (5, 3) to carry secret messages ($s_j$) in Fig. 2. Then, some valid candidate pixel pairs whose shape is the same as the pre-determined LT-shape are searched, e.g., $RM(7, 3)$, $RM(4, 1)$ and $RM(2, 5)$. Finally, the $RM(7, 3)$ is chosen because it has the shortest distance to $RM(5, 3)$. In other words, the original pixel pair (5, 3) is changed to (7, 3) in the stego-image and two secret messages (01 00) have been hidden.

### 2.3 Pazhitnov’s Tetris puzzle game

Tetris [25, 30] was invented by Alexey Pazhitnov in 1985, and it is one of most popular computer games. Tetris is a tetromino-based falling puzzle game whose objective is to form some solid horizontal lines without gaps by rotating and sliding tetrominoes within an $M \times N$ gameboard, and then the filled line disappears. Once a line is filled and disappears, the player gains points. A tetromino is a geometric shape composed of four grid-squares connected orthogonally. In general, there are seven types of tetrominoes in total in a Tetris game, including ‘T’ (T), ‘Square’ (SQ), ‘Right Snake’ (RS), ‘Left Snake’ (LS), ‘Right Gun’ (RG), ‘Left Gun’ (LG), and ‘I’ (I), as shown in Fig. 3a. The allowed rotation angles $\theta$ include four types: [0°, 90°, 180°, 270°]. Take tetromino ‘T’ for example. Assume that the first shape ‘T’ shown in Fig. 3b is a base tetromino with the orientation of $\theta = 0°$, and three other shapes of derivatives with three orientations of 90°, 180°, and 270°, are depicted on its right side, respectively. To sum up, Tetris includes seven types of tetrominoes in total, and each tetromino can be derived into different shapes with each shape also consisting of four grid-squares. Concretely, in our approach, a series of combinations of the tetrominoes and their derivatives will be conducted and then utilised to fill an $M \times N$ gameboard without gaps to generate a meaningful matrix, which will be adopted as the fundamental framework of the reference matrix in our proposed DH scheme.

### 3 Proposed scheme

Inspired by Liu et al.’s location-based DH scheme [14], in this paper, we present a novel DH scheme based on Tetris to further enhance the image quality of stego-image and EC. The proposed scheme mainly consists of two phases: (i) secret messages embedding phase; and (ii) secret messages extraction phase. Before that, we shall introduce how to construct the reference matrix based on Tetris, called $MT$, and the corresponding LUT in Sections 3.1 and 3.2, respectively, for later use in the data embedding phase.

#### 3.1 Reference matrix construction

Similar to Sudoku-based DH scheme, in order to effectively embed secret messages, a Tetris-based reference matrix, called $MT$, with a size of 256 × 256, should be first constructed. In our proposed scheme, the structure of the $MT$ is quite novel, and the construction of $MT$ is like to play a jigsaw puzzle game.

We take a square lattice, $Q$, with the size of $L_Q \times L_Q$. Obviously, there are $L_Q^2$ empty grid-squares in the $Q$. Although the selection of $L_Q$ has various possibilities, it should meet this rule, i.e. the $L_Q^2$ must be divisible by 4 (Any one of tetrominoes has 4 grid-squares), making those $L_Q^2/4$ tetrominoes can just appropriately fill out the corresponding square $Q$ without gaps. As a result, values of $L_Q$ can be determined, and are ranged in [2, 4, 6, 8, 10, ..., 256]. Next, for simplicity, we mainly discuss the cases when $L_Q$ is 4 and 6, which can lead to a considerable EC, in this paper. After a $Q$ has been prepared, a jigsaw puzzle game is conducted in an entire $Q$ and its objective is to make all grid-squares are filled without gaps by rotating and sliding some tetrominoes while making each tetromino has different shape. It is certain that various splicing solutions of a tetromino can be derived into different shapes with each shape also consisting of four grid-squares. Concretely, in our approach, a series of combinations of the tetrominoes and their derivatives will be conducted and then utilised to fill an $M \times N$ gameboard without gaps to generate a meaningful matrix, which will be adopted as the fundamental framework of the reference matrix in our proposed DH scheme.

#### 3.2 Proposed DH scheme

In this section, we will introduce our proposed DH scheme based on Tetris, called $MT$, and the corresponding LUT in Sections 3.1 and 3.2, respectively, for later use in the data embedding phase.

First, the tetrominoes $T_1$, $T_2$, $T_3$, and $T_4$ are shown in Fig. 4a. Next, for simplicity, we mainly discuss the cases when $L_Q$ is 4 and 6, which can lead to a considerable EC, in this paper. After a $Q$ has been prepared, a jigsaw puzzle game is conducted in an entire $Q$ and its objective is to make all grid-squares are filled without gaps by rotating and sliding some tetrominoes while making each tetromino has different shape. It is certain that various splicing solutions of a tetromino can be derived into different shapes with each shape also consisting of four grid-squares. Concretely, in our approach, a series of combinations of the tetrominoes and their derivatives will be conducted and then utilised to fill an $M \times N$ gameboard without gaps to generate a meaningful matrix, which will be adopted as the fundamental framework of the reference matrix in our proposed DH scheme.
secret messages. In addition, each shape also has four grid squares and is always filled in four distinct digits ranging in [0, 3], as shown in Figs. 4b and 5b. Similarly, each digit can represent a two-bit secret message later.

After the spliced Q has been generated, the MT with a size of 256 × 256 is constructed by replicating this pre-decided Q. Fig. 6 shows examples of the corresponding reference matrices MT4 and MT6, which are derived from Q4 and Q6, respectively.

Fig. 6 demonstrates the following features: (i) In MT4, in any 4 × 4 square, such as the white dotted rectangles shown, four different shapes of tetrominoes, i.e. ‘T1’, ‘T2’, ‘T3’, and ‘T4’, can always be found. (ii) Similarly, nine shapes of tetrominoes, i.e. ‘T1’, ‘T2’, ‘T3’, ‘T4’, ‘SQ’, ‘RS1’, ‘RS2’, ‘LS1’, and ‘LS2’ can always be found for any given 6 × 6 square in MT6. Such unique feature guarantees that given a coordinate in MT4, a 4 × 4 square can be determined; then, four different shapes of tetrominoes within the square can always be found. A similar feature also exists in MT6, but the type of distinct tetromino shapes is changed to nine. Based on this feature, we will construct a mapping relation between tetrominoes’ shapes and secret digits (see Fig. 7) and then use it to serve for concealing secret messages.

3.2 LUT construction

After Q and MT have been prepared, the corresponding LUT based on tetrominoes shall be constructed for data embedding. The construction of LUT can be processed by the rules as follows: (i) Scan all shapes of used tetrominoes within Q. (ii) Sort those tetrominoes first by the order in which they appear in Fig. 3a. (iii) For the same kind of different shapes of tetrominoes, sort them by the ascending order of the rotation angle. (iv) Number this sorted sequence using the shape number from 0 to (L\(^2\)Q\(^4\))/4! – 1. Finally, the LUT can be derived, such as LUT4 and LUT6, as shown in Fig. 7.

As Fig. 7a shows, LUT4 is designed for MT4 to carry a 4-bit of secret message ranging from (‘00 00’) to (‘11 11’). In LUT4, each column includes four distinct shapes of tetrominoes, which can represent four distinct secret digits ranging from 0 to 3 and their corresponding binary representations are mapped to [‘00’, ‘01’, ‘10’, ‘11’], which were listed in the left of LUT4. Meanwhile, each row has the same tetromino, where its four grid-squares separately cover four different secret digits ranging from 0 (‘00’) to 3 (‘11’), which were listed in the bottom of LUT4. For instance, the filled square (called an LUT-shape) located at (2, 2) in LUT4 means the following: (i) this LUT-shape belongs to ‘T3’ (see Fig. 4b), which represents a binary code ‘10’ in terms of its shape number; (ii) this LUT-shape contains a value of 2, which also maps to a binary code ‘10’. Based on above two facts, it is concluded that the LUT-shape can be used to carry a 4-bit binary stream (‘10 10’).

Similarly, we also designed the LUT6 to carry (log9 + 2) bits of secret messages. In LUT6, each column includes nine different shapes of tetrominoes, which can be used to conceal (log9) -bit of secret messages, and its corresponding digits and binary codes were listed in the left of LUT6. Also, each row covers four different digits separately filled in different grid-squares of a tetromino, which is able to carry 2-bit of secret messages. In other words, for example, when the LUT-shape is encountered, we can look for it in LUT6, and then its location is finally determined as (6, 2). Therefore, it can be used to carry secret bits (‘110 10’), which its binary code maps to.

When the MT and LUT have been prepared, the data embedding and extraction can be processed. Assume that secret messages need to be embedded into a pixel pair \((p_i, p_{i+1})\). First, the corresponding LUT-shape must be located by projecting the secret messages into the LUT; then, the pixel pair \((p_i, p_{i+1})\) is changed to the other pixel pair so that the new shape and shape's value are the same as the determined LUT-shape. During the process of concealing secret messages into a pixel pair, there are multiple candidates of pixel pairs that can be found by combining LUT and MT. Finally, a candidate pixel pair \((p_i', p_{i+1}')\) which causes the least distortion with the original pixel pair \((p_i, p_{i+1})\) is chosen. The distortion \(d\) is measured by:

\[
\text{distortion } d = \frac{1}{N} \sum_{i=1}^{N} |p_i - p_i'|^2
\]
In addition, it is worth noting that the usable spliced \( Q \) should be shared between the sender and the recipient in advance for the aim of extracting the secret messages correctly. With the shared spliced \( Q \), the reference matrix \( MT \) and LUT can be correctly derived.

### 3.3 Secret message embedding phase

For a greyscale image \( I \) with the size of \( H \times W \), all pixels are paired by a pairing technique. For simplicity, the paired results are denoted as \( I = \{ (p_i, p_{i+1}) \} \). At the same time, assume that the secret messages are \( B \) and its length is \( L_B \). According to the property that the selection of \( MT \) is classified into two cases. (i) If \( L_B \leq 2 \cdot H \cdot W \), the \( MT_4 \) is chosen to serve for DH to guarantee the visual quality of stego-image as high as possible. In this case, the secret messages are converted into the secret digits in the base-9 numeral system. The converted secret digits are further paired sequentially and denoted as \( S = \{ (s_i, s_{i+1}) \} \), where both the \( s_i \) and \( s_{i+1} \) are the base-9 digits. (ii) If \( 2 \cdot H \cdot W < L_B \leq (\log_9(9 + 2) \cdot H \cdot W)/2 \), the \( MT_6 \) is selected to achieve higher EC along with satisfactory stego-image quality. Correspondingly, we convert the secret messages into secret digits in the base-9 and base-4 numeral systems in turn. Concretely, the first several (3 or 4) bits of secret messages are converted into a base-9 digit firstly and then the following 2 bits are converted into a base-4 digit secondly. This process is repeated until all secret messages have been handled. For convenience, the converted secret digits are denoted as \( S = \{ (s_i, s_{i+1}) \} \) for case 2, and the major difference from case (1) is that \( s_i \) is a base-9 digit while \( s_{i+1} \) is a base-4 digit in case 2.

Next, in order to embed secret digits \((s_1, s_{i+1})\) into an original pixel pair \((p_i, p_{i+1})\), we can process it by the following rules. Firstly, an LUT-shape in LUT is determined according to the to-be-embedded secret digits \((s_1, s_{i+1})\), where the \( s_1 \) and \( s_{i+1} \) are treated as the row and column index values in LUT, respectively. Secondly, the pixel pair \((p_i, p_{i+1})\) is projected into the selected \( MT \) and denoted as \( MT(p_i, p_{i+1}) \), where the \( p_i \) is row index value and the \( p_{i+1} \) is the column index value. By the properties of \( MT \) introduced above, it is certain that in the adjacent areas of \( MT(p_i, p_{i+1}) \), we can find multiple candidates \( MT(p', p'_{i+1}) \) whose tetromino location is with the same shape as the determined LUT-shape and its projected digit is equal to the value of the determined LUT-shape. Finally, an \((p', p'_{i+1})\) that has the least distortion with the original pixel pair \((p_i, p_{i+1})\) is selected as the final pixel pair of the stego-image. In other words, the pair \((p', p'_{i+1})\) is changed into pair \((p''_i, p''_{i+1})\) to carry the secret digits \((s_i, s_{i+1})\). After the embedding procedure, the stego-image SI is sent to the recipient.

In order to demonstrate the embedding procedure more clearly, two examples regarding secret digits embedding using \( MT_4 \) and \( MT_6 \) are illustrated in Figs. 8 and 9, respectively.

#### Examples for \( MT_4 \)

**Case 1:** Assume an original pixel pair is \((0, 8)\), the secret messages are ‘01 00’ and its 4-ary secret digits are \((s_1, s_{i+1})=(1, 0)\). Using the \( s_1 = 1 \) as the row index value of \( LUT_4 \) and the \( s_{i+1} = 0 \) as the row index value of \( LUT_4 \).
column index value; thus, the corresponding LUT-shape is determined. Then, we do a search in $MT_6$ to find multiple candidate pixel pairs whose tetromino has the same shape (‘T’), so the secret messages are ‘1000 00’. In this case, the first three secret messages ‘1000’ are converted into a 9-ary digit $s_t$ and the remaining two secret messages are transferred into a 4-ary digit $s_{t+1}$, and hence it results as $(s_t, s_{t+1}) = (4, 2)$. To begin with, we think of the $s_t$ as a row index value and $s_{t+1}$ as a column index value and then project them in LUT$_6$. Finally, an LUT-shape located at LUT$_6(4, 2)$ is found. Afterwards, a series of candidates, e.g. $MT_6(3, 3)$, $MT_6(3, 9)$ and $MT_6(9, 3)$, as red rhombuses shown in Fig. 9, are looked for. Then, the distance between $(p_i, p_{i+1})$ and the candidates is measured to select a candidate that causes the least distortion. Obviously, the $MT_6(3, 3)$ is the closest one of these, so the original pixel pair $(1, 1)$ is modified to stego-pixel pair $(p'_i, p'_{i+1}) = (3, 3)$. Case 2: Take another specific case for example. Suppose the secret messages are ‘1000 00’ and its corresponding converted secret digits $(s_t, s_{t+1})$ are $(8, 0)$. They should be concealed into the pixel pair $(6, 7)$. Certainly, we use $(s_t, s_{t+1}) = (8, 0)$ to find an LUT-shape in LUT$_6$ in the same way, i.e. ‘LS’. From Fig. 9, we observe that the $MT_6(6, 8)$ satisfies all of the requirements: (i) its located tetromino belongs to ‘LS’; (ii) its value is equal to $s_{t+1} = 0$; and (iii) it is the nearest one to the $MT_6(6, 7)$. Therefore, the pixel pair $(6, 7)$ is changed to (6, 8). It means that the pixel pair (6, 8) in the stego-image carries secret messages ‘1000 00’.

3.4 Secret message extraction phase

When the recipient obtains the stego-image $SI$ and the spliced $Q$, the LUT can be reconstructed in the same way mentioned in Section 3.2, and then the process of data extraction can be performed. Firstly, s/he constructs the reference matrix $MT$ using the spliced $Q$. Secondly, the stego-image $SI$ is divided into non-overlapping pixel pairs in the same way and is denoted as $SI = \{(p_i, p_{i+1})\mid i = 1, 3, 5, 7, \ldots, (H \cdot W-1)\}$. To extract the secret messages, the recipient projects each stego-pixel pair $(p'_i, p'_{i+1})$ into the $MT$, i.e. $MT(p'_i, p'_{i+1})$. Then, an LUT-shape can be determined by combining the shape of the tetromino the $MT(p'_i, p'_{i+1})$ locates on and the value of $MT(p'_i, p'_{i+1})$. According to the aforementioned determined LUT-shape, we can find its corresponding coordinate $(s_t, s_{t+1})$ in LUT, those found coordinate shall be the secret digits. Finally, those secret digits are converted into binary codes. After concentrating all derived binary codes, the original secret messages are generated. The same operations are performed until all pixel pairs are processed and the secret messages can be extracted.

Examples: For simplicity, we only take two examples for $MT_6$ and LUT$_6$ to explain the process of secret message extraction.

Case 1: Assume that the $(p_i, p_{i+1})$ is (1, 1) and a segment of secret messages ‘100 10’. In this case, the first three secret messages ‘100’ are converted into a 9-ary digit $s_t$ and the remaining two secret messages are transferred into a 4-ary digit $s_{t+1}$, and hence it results as $(s_t, s_{t+1}) = (4, 2)$. To begin with, we think of the $s_t$ as a row index value and $s_{t+1}$ as a column index value and then project them in LUT$_6$. Finally, an LUT-shape located at LUT$_6(4, 2)$ is found. Afterwards, a series of candidates, e.g. $MT_6(3, 3)$, $MT_6(3, 9)$ and $MT_6(9, 3)$, as red rhombuses shown in Fig. 9, are looked for. Then, the distance between $(p_i, p_{i+1})$ and the candidates is measured to select a candidate that causes the least distortion. Obviously, the $MT_6(3, 3)$ is the closest one of these, so the original pixel pair $(1, 1)$ is modified to stego-pixel pair $(p'_i, p'_{i+1}) = (3, 3)$. Case 2: Take another specific case for example. Suppose the secret messages are ‘1000 00’ and its corresponding converted secret digits $(s_t, s_{t+1})$ are $(8, 0)$. They should be concealed into the pixel pair $(6, 7)$. Certainly, we use $(s_t, s_{t+1}) = (8, 0)$ to find an LUT-shape in LUT$_6$ in the same way, i.e. ‘LS’. From Fig. 9, we observe that the $MT_6(6, 8)$ satisfies all of the requirements: (i) its located tetromino belongs to ‘LS’; (ii) its value is equal to $s_{t+1} = 0$; and (iii) it is the nearest one to the $MT_6(6, 7)$. Therefore, the pixel pair $(6, 7)$ is changed to (6, 8). It means that the pixel pair (6, 8) in the stego-image carries secret messages ‘1000 00’.

Fig. 8 Example of secret messages embedding using $MT_4$

Fig. 9 Example of secret messages embedding using $MT_6$
4 Experimental results

In this section, experiments were implemented using MATLAB 2019a on a personal computer with an Intel® Core (TM) i5-1035G1 CPU @ 1.00 GHz 1.19 GHz, 16 GB RAM, to demonstrate the efficacy of the proposed scheme. Eight 8-bit depth greyscale images with a size of 512 × 512, demonstrated in Fig. 10, were served as test images. Besides, in our experiments, the binary secret messages were randomly generated by using a seed, except for the discussion of the specific sequence of secret messages as shown in Table 3.

In our experiments, several statistical metrics are computed for the performance evaluation. Firstly, the PSNR [31] is utilised to evaluate the visual quality of the stego-image and is defined by:

\[
\text{PSNR} = 10 \log_{10} \left( \frac{255 \cdot H \cdot W}{\sum_{i=1}^{H} \sum_{j=1}^{W} (I_{i,j} - \bar{I}_{i,j})^2} \right)
\]  

(2)

where the \(I_{i,j}\) and \(\bar{I}_{i,j}\) represent the pixel values of the original image \(I\) and the stego-image \(\bar{I}\), respectively. Generally, the higher the PSNR is, the better the quality of the image is. As long as PSNR is larger than 30 dB, the human vision system is hard to distinguish the stego-image from the original image.

Secondly, both the EC and ER are used to measure the ability of the stego-image to carry secret messages and is defined as (3)

\[
\text{EC} = L_B, \quad \text{ER} = \frac{\text{EC}}{H \cdot W}.
\]  

(3)

where \(L_B\) (or EC) represents the length of embedded secret messages.

Thirdly, we employ the SSIM (structural similarity) [32] to evaluate the change of the structure of stego-image. It is a measure that can estimate the similarity between two images with respect to the structure and is ranged in \([0, 1]\). SSIM considers image degradation as a perceived change in structural information and can be defined as the following:

\[
\text{SSIM}(I, \bar{I}) = \left( \frac{2 \cdot \mu_I \cdot \mu_{\bar{I}} + c_1}{\mu_I^2 + \mu_{\bar{I}}^2 + c_1} \right) \cdot \left( \frac{2 \cdot \sigma_{I,\bar{I}} + c_2}{\sigma_I^2 + \sigma_{\bar{I}}^2 + c_2} \right).
\]  

(4)

where \(\mu_I\) and \(\mu_{\bar{I}}\) are the mean of the images \(I\) and \(\bar{I}\), respectively; \(\sigma_I^2\) and \(\sigma_{\bar{I}}^2\) are the variance of the images \(I\) and \(\bar{I}\), respectively. \(\sigma_{I,\bar{I}}\) is the covariance of \(I\) and \(\bar{I}\). \(c_1 = (k_1 \cdot L)^2\), \(c_2 = (k_2 \cdot L)^2\) are two variables to stabilise the division with the weak denominator, where \(L\) is the dynamic range of the pixel values and \(k_i (i = 1, 2)\) is a constant much <1. The closer to 1 the SSIM index, the higher the similarity is.

Finally, NCC (normal cross correlation) [33], as another method for measuring the similarity of two data sequences, is also measured in our experiments. It ranges from 0 to 1 and is defined by:

\[
\text{NCC}(I, \bar{I}) = \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} (I_{i,j} - \mu_I) \cdot (\bar{I}_{i,j} - \mu_{\bar{I}})}{\sqrt{\sum_{i=1}^{H} \sum_{j=1}^{W} (I_{i,j} - \mu_I)^2} \cdot \sqrt{\sum_{i=1}^{H} \sum_{j=1}^{W} (\bar{I}_{i,j} - \mu_{\bar{I}})^2}}.
\]  

(5)

If NCC = 1, it indicates two images are the same.

4.1 Results of our approach

Fig. 11 shows the stego-images when the maximum ER of 2.56 bpp was achieved. It is clear that the visual quality of the stego-image remains good even if the embedded secret messages are up to 669,992 bits. So, it is hard for the human eye to detect any distortion caused in the images. Tables 1 and 2 list the performances of our approach when \(L_Q\) is set to 4 and 6, respectively. Note that the sequence of secret messages is randomly generated. When \(L_Q = 4\), the average ER was 2.0 bpp, and the average PSNR of the stego-images was 46.38 dB. We also can see that the corresponding SSIM and NCC were 0.9977 and 0.9997, respectively, both of which imply that the similarity between the original image and the stego-images is very high. When \(L_Q = 6\), our proposed scheme had a higher ER of 2.56 bpp, and the stego-image's PSNR was around 43.12 dB. It is also obvious to see that the SSIM and NCC of this scheme remains a high value, 0.9960 and 0.9994, respectively. This indicates that the stego-image’s structure remains good even when the ER is up to 2.56 bpp. Furthermore, the last column of Tables 1 and 2 show the BER (bit error rate) of the extracted secret messages, with values of 0, indicating that the proposed scheme can extract the secret messages with error-free.

Additionally, we also experimented the proposed scheme on four specific sequences of secret messages, which are quite different from the aforementioned random secret messages. Those four specific sequences of secret messages are: (i) all 0's; (ii) all 1's; (iii) a sequence of '01010101…'; (iv) a sequence of '10101010…'. The results are demonstrated in Table 3. As we can see, no matter ER is set to 2.0 or 2.56 bpp, the performances tested on specific secret messages is almost the same as that tested on random secret messages, with respect to the PSNR, SSIM, and NCC. Hence, it is concluded that the visual quality and structure of stego-images also keep well with our proposed scheme based on our experimental data. Besides, those specific secret messages also can be extracted losslessly, thus, resulting in BER = 0.

4.2 Security analysis

While achieving high EC, our approach also provides secure covert communication. To prove this point, more detailed, security analyses like PVD (pixel value difference) histogram [34], RS steganalysis [35] and relative entropy (RE) [36] are employed to analyse the stego-images with an ER of 2.56 bpp. In the security
analysis, eight greyscale images were explored in our experiments. However, only two different image sets are selected and demonstrated in the following subsections to show the generality of our proposed scheme.

4.2.1 PVD histogram: The PVD histogram is a method for examining the degree of differences in the neighbouring pixels between the original image and the stego-image. According to the conclusions of the PVD histogram analysis in [34], the abnormal behaviour of the PVD histogram reveals the presence of the hidden messages. It is also possible to estimate the size of the embedded secret messages. Fig. 12 shows that the PVD histogram curves of the ‘Baboon’ and ‘Peppers’ images embedded with secret messages were well preserved, which is quite similar to those of the original images without carrying secret messages.

4.2.2 RS steganalysis: We also analyse the performance of our approach in terms of resisting the RS steganalysis. The RS steganalysis is based on the following functions: a discrimination function $DF$, a flipping function $F^+$ and a shifting function $F^-$. The divided image blocks are separated into three groups, including $R$ (Regular), $S$ (Singular) and $U$ (Unchanged), by combining those

![Fig. 11 Stego-images. Top row (left to right): Baboon(43.13 dB), Boat(43.11 dB), Elaine(43.10 dB), and Goldhill(43.14 dB). Bottom row (left to right): Lake(43.12 dB), Lena(43.13 dB), Peppers(43.13 dB), and Airplane(43.13 dB)](image)

| Images   | ER, bpp | PSNR, dB | SSIM | NCC  | BER, bits |
|----------|---------|----------|------|------|-----------|
| Baboon   | 2.0     | 46.37    | 0.9987 | 0.9995 | 0         |
| Boat     | 2.0     | 46.37    | 0.9979 | 0.9979 | 0         |
| Elaine   | 2.0     | 46.36    | 0.9974 | 0.9974 | 0         |
| Goldhill | 2.0     | 46.39    | 0.9982 | 0.9982 | 0         |
| Lake     | 2.0     | 46.37    | 0.9978 | 0.9978 | 0         |
| Lena     | 2.0     | 46.39    | 0.9973 | 0.9973 | 0         |
| Peppers  | 2.0     | 46.38    | 0.9972 | 0.9972 | 0         |
| Airplane | 2.0     | 46.38    | 0.9967 | 0.9967 | 0         |
| Average  | 2.0     | 46.38    | 0.9977 | 0.9977 | 0         |

| Images   | ER, bpp | PSNR, dB | SSIM | NCC  | BER, bits |
|----------|---------|----------|------|------|-----------|
| Baboon   | 2.56    | 43.13    | 0.9978 | 0.9989 | 0         |
| Boat     | 2.56    | 43.11    | 0.9964 | 0.9963 | 0         |
| Elaine   | 2.56    | 43.10    | 0.9957 | 0.9957 | 0         |
| Goldhill | 2.56    | 43.14    | 0.9969 | 0.9969 | 0         |
| Lake     | 2.56    | 43.12    | 0.9963 | 0.9963 | 0         |
| Lena     | 2.56    | 43.13    | 0.9953 | 0.9953 | 0         |
| Peppers  | 2.56    | 43.13    | 0.9953 | 0.9953 | 0         |
| Airplane | 2.56    | 43.13    | 0.9945 | 0.9945 | 0         |
| Average  | 2.56    | 43.12    | 0.9960 | 0.9960 | 0         |

| Type of messages | ER = 2.0 bpp | ER = 2.56 bpp |
|------------------|--------------|--------------|
|                  | EC, bits    | PSNR, dB    | SSIM | NCC  | BER, bits | EC, bits    | PSNR, dB    | SSIM | NCC  | BER, bits |
| i                 | 524,288     | 46.36        | 0.9976 | 0.9997 | 0         | 655,360     | 43.13        | 0.9959 | 0.9993 | 0         |
| ii                | 524,288     | 46.36        | 0.9976 | 0.9997 | 0         | 655,360     | 43.12        | 0.9959 | 0.9993 | 0         |
| iii               | 524,288     | 46.37        | 0.9977 | 0.9997 | 0         | 655,360     | 43.12        | 0.9960 | 0.9993 | 0         |
| iv                | 524,288     | 46.38        | 0.9976 | 0.9997 | 0         | 655,360     | 43.12        | 0.9960 | 0.9993 | 0         |
| Average           | 524,288     | 46.37        | 0.9976 | 0.9997 | 0         | 655,360     | 43.12        | 0.9960 | 0.9993 | 0         |
functions with masks $M$ and $-M$. In detail, we define the parameters $R_M$ and $S_M$ that represent the percentage of $R$ and $S$ groups when the functions and $M$ were applied. Similarly, $R_{-M}$ and $S_{-M}$ are parameters that separately represent the percentages of $R$ and $S$ groups when the corresponding functions and $-M$ were applied. If a data embedding algorithm resists the RS test, it should satisfy the following equation:

$$R_M \cong R_{-M} \quad \text{and} \quad S_M \cong S_{-M}.$$  (6)

Fig. 13 shows the graphs of the RS steganalysis for the stego-images produced for two test images ‘Lena’ and ‘Peppers’. As we can see from the graphs, the gaps between $R_M$ and $R_{-M}$, and between $S_M$ and $S_{-M}$ curves are very close to each other. Obviously, (6) is satisfied in the stego-images ‘Lena’ and ‘Peppers’ with the proposed scheme. This implies that our approach is robust against the RS steganalysis.

4.2.3 Relative entropy: Furthermore, the statistics of entropy and relative entropy were introduced to evaluate the divergence of the stego-image ($SI$) from the original image ($I$). Relative entropy is an indicator that can be used to measure the difference between probability distributions of pixel values for two images, i.e. $I(p)$ and $SI(p)$, can be defined by:

$$\text{RE}(I, SI) = \sum_{p=0}^{255} I(p) \cdot \frac{I(p)}{SI(p)}.$$  (7)

When $\text{RE} = 0$, two images are a coincidence and the system is perfectly secure. Table 4 lists the experimental results of entropy and RE for all test images. We can observe that the entropy values for the image $I$ and the stego-image $SI$ are extremely close to each other, with an average difference of 0.0103, while the corresponding value of RE between them is about 0.004, which is also close to zero. It is concluded that our approach is quite secure.

To sum up, three security analyses confirm that the difference between the original image and stego-images are quite small, and no clue can be found with our proposed scheme.

4.3 Comparison and analysis

4.3.1 Comparisons in performance: To further demonstrate the superior performance of our proposed Tetris-based scheme, we compared the results provided by the proposed scheme with the results provided by other previous works [14, 16–21, 26–29, 37–39].

Firstly, a performance comparison of our proposed scheme, a magic cube-based DH scheme [19], and five representative TS-based DH schemes [14, 16–18, 39] is conducted for six test images and is shown in Table 5. From Table 5, we can see that the average ER (or EC) of our approach is the same as that of schemes [16, 18, 39] and is higher than schemes [14, 17, 19] with differences of 0.5,
0.15, and 0.25 bpp, respectively. Moreover, our approach provides a better stego-image visual quality than that of [16, 18, 39] under the same ER, with the differences of 1.35, 1.50, and 2.21 dB, respectively.

Secondly, the proposed scheme is also compared to the Sudoku-based schemes [20, 21, 26–29], with respects to the EC, ER and image quality of the four test images. Those schemes are all designed on the concept of a puzzle game. The results are presented in Table 6. It is apparent that the proposed Tetris-based scheme has the highest PSNR value when compared to schemes [20, 26–29], where ER reaches 1.5 or 2.0 bpp. When EC is set to 414,188 bits, the PSNR of our approach is slightly less than that of Hong et al.’s scheme [21]. However, based on the reports presented in [21], their EC is limited to the ER of 1.58 bpp. By contrast, our approach promotes an ER up to 2.56 bpp while maintaining the average PSNR of 43.12 dB. Therefore, our proposed scheme is an excellent one that obtains both a higher EC and a higher PSNR.

Finally, we further provide a summary and comparison of PSNR and ER between the proposed Tetris-based scheme and other recent works [14, 16, 18, 20, 26–28, 37, 38], as shown in Table 7. Herein, ‘-’ represents unavailable or unspecific. In Table 7, it is easy to find that the PSNR of our approach maintains a considerable level when ER is 1.5 and 2.0 bpp, that is, 48.72 and 46.38 dB, respectively. It is also clear that given the larger ER of 2.5 or 3.0 bpp, the visual quality of our proposed scheme’s stego-image reaches 43.22 and 40.72 dB, respectively. The Sudoku-based DH schemes, like [20], have a weakness of the limitation of EC. For the TS-based schemes, such as Leng [37], their payload and image quality depend on the width and height of both the octagon and the corner of the octagon. As a result, the PSNR seriously decreases as the octagon is extended, e.g. a PSNR of 39.33 dB is lower than that of our approach when the ER reaches 3.0 bpp. Similarly, Xie et al. [38] presented a revised version based on the EMD-2 scheme by constructing an extended squared magic matrix. Although their scheme can achieve a larger ER high up to 3.15 bpp, it is relatively poor in the visual quality of the stego-image, with an average PSNR of 39.89 dB (ER = 3.0 bpp). Only our approach, Li et al.’s scheme [18] and Chen et al.’s scheme [28] have both a larger ER up to 3.0 bpp and a satisfactory PSNR not lower than 40.00 dB. Among which, the PSNR of our approach is ranked second after that of Li et al.’s scheme [18] when ER reaches 3.0 bpp. That is because, theoretically, the maximum distortion of 128 in our approach is a little greater than that of 78 in the scheme [18] at a time. However, compared to our approach, Li et al.’s scheme [18] and Chen et al.’s scheme [28] have a weakness of visual quality when the ER was set as 1.5, 2.0 and 2.5 bpp. In
summary, it is enough to see that the proposed Tetris-based DH scheme significantly outperforms other state-of-the-art schemes [14, 16, 18, 20, 26–28, 37, 38].

4.3.2 Comparisons in various features: To indicate the differences of DH schemes more intuitively, comparisons of features for them are given in Table 8. The detailed discussions are listed as follows:

- Compare with LSB matching scheme [9]. The pre-shared knowledge, including reference matrix or LUT or location table, is required for secret messages embedding and extraction in our approach and other DH schemes [13–22, 26–29], whereas scheme [19] does not need it. Another difference is that the ERs of our approach and other DH schemes [13–22, 26–29] are significantly higher than [9].

- Compare with TS or extended-TS-based DH schemes [13–18] and 3D magic cube-based DH scheme [19]. On the one hand, both those schemes and our approach needed to pre-share the rules of constructing reference matrix or LUT or location table between sender and receiver. On the other hand, our approach achieved a higher ER while maintaining a better visual quality of stego-image. Moreover, the maximum ER of our approach reaches 3.0 bpp, which is much greater than their maximum ER of 2.5 bpp.

- Compare with Sudoku or mini-Sudoku-based DH schemes [20–22, 26–29]. Firstly, similar to our approach, the pre-shared knowledge is also required in those schemes. Besides, it is undeniable that the Sudoku-based schemes provide relatively reliable security because of the hundreds of millions of fundamental possible solutions of Sudoku. On the aspects of ER and PSNR, both our approach and scheme [28] obtain the maximum ER of 3.0 bpp, however, the PSNR of stego-image provided by our approach is 0.71 dB which is higher than that of scheme [28].

- Compare with Jigsaw- and Tetris-based schemes [24, 25]. Although there is no requirement in sharing knowledge in advance, the ER and PSNR provided by schemes [24, 25] are far lower than those of our approach. Concretely, they hide secret messages with the use of shapes of Jigsaw or Tetris, thus, resulting in a relatively low ER. Meanwhile, scheme [24] provided an unsatisfactory visual quality of stego-image, with a PSNR of 34.00 dB.

4.3.3 Computational cost analysis: In our approach, the required computational cost mainly lies in five aspects: construct the reference matrix, construct the LUT, divide pixel pairs, embed secret messages and extract secret messages. Table 9 lists a comparison of the computational cost, and details are analysed as below: the computational complexity in constructing reference matrix; 2 LUT or location table; 3 None; 2D Reference matrix is a structure of two-dimensional; 3D Reference matrix is a structure of three-dimensional; N/A Not applicable; N/P Not present.

| Schemes                  | PSNR, dB |
|-------------------------|----------|
| Chang et al. [20]       | 44.90    |
| Lyu et al. [26]         | 47.51    |
| Liu et al. [14]         | 49.95    |
| He et al. [27]          | 48.06    |
| Liu et al. [16]         | 52.33    |
| Leng et al. [37]        | 49.74    |
| Xie et al. [38]         | 42.90    |
| Li et al. [18]          | 43.94    |
| Chen et al. [28]        | 48.14    |
| Proposed scheme         | 48.72    |

Table 8 Comparison of features for different DH schemes

| DH schemes           | Methodology | Dimension | Preshared knowledge | ER, bpp | PSNR, dB |
|----------------------|-------------|-----------|---------------------|--------|----------|
| Mielikainen [9]      | LSB         | 2D        | 3                   | 1.00   | N/P      |
| Chang et al. [13]    | TS          | 2D        | 3                   | 1.50   | 49.40    |
| Liu et al. [14]      | TS          | 2D        | 2                   | 2.00   | 45.55    |
| Jin et al. [15]      | TS          | 2D        | 1                   | 2.00   | 45.56    |
| Liu et al. [16]      | Extended-TS | 2D        | 1                   | 2.50   | 41.87    |
| Liu et al. [17]      | Extended-TS | 2D        | 1                   | 2.35   | 44.31    |
| Li et al. [18]       | Extended-TS | 2D        | 1                   | 2.50   | 41.72    |
| Lee et al. [19]      | Magic cube  | 3D        | 1, 2                | 2.25   | 44.00    |
| Chang et al. [20]    | Sudoku      | 2D        | 1                   | 1.50   | 44.90    |
| Hong et al. [21]     | Sudoku      | 2D        | 1                   | 1.58   | 47.49    |
| Hong et al. [22]     | Sudoku      | 2D        | 1                   | 1.58   | 48.67    |
| Lyu et al. [26]      | Sudoku      | 2D        | 1                   | 1.50   | 47.51    |
| He et al. [27]       | Mini-Sudoku | 2D        | 1, 2                | 2.00   | 46.37    |
| Chen et al. [28]     | Mini-Sudoku | 2D        | 1, 2                | 3.00   | 40.01    |
| Horng et al. [29]    | Mini-Sudoku | 3D        | 1, 2                | 2.00   | 46.37    |
| Farn and Chen [24]   | Jigsaw      | 2D        | 9928–10242         | 34.00  |
| Ou and Chen [25]     | Tetris      | 2D        | 3                   | 3.00   | 40.72    |

1 Reference matrix; 2 LUT or location table; 3 None; 2D Reference matrix is a structure of two-dimensional; 3D Reference matrix is a structure of three-dimensional; N/A Not applicable; N/P Not present.

IET Image Process., 2020, Vol. 14 Iss. 17, pp. 4633-4645
This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)
4.3.4 Potential failure analysis: with an overall execution time of 0.591 s. Exceeding the performance of some state-of-the-art schemes in this manner, we can prove that the proposed scheme has excellent performance, as noted in Table 10. As both EC and visual quality are considered, it is essential to assess the computational complexity in embedding or extracting secret messages. The overall computational complexity of our approach can be represented as $O(H/W) = O(L^2 Q/2)$.

Considering the $L_Q$ is usually far smaller than $H$, thus, the overall computational complexity of our approach can be represented by $O(H/W) = O(L^2 Q/2)$. By the same way, the computational complexity of scheme [14] is $O(H/W) = O(L^2 Q/4)$ because of the required processing of embedding secret messages with the help of a 4 × 4 location table. Similarly, the computational complexity of scheme [18] is $O(H/W) = O(L^2 Q/4)$, where the LUT is sized 4 × 16. For scheme [28], due to the three-layer mini-Sudoku designed to serve for secret messages embedding or extraction, its computational complexity is as $O(H/W) = O(L^2 Q/4)$. Additionally, it is also worth noting that the temporary storage cost of reference matrix for those four schemes are all sized of 256 × 256, whereas the storage cost of the LUT is sized of 4 × 4, 4 × 16, 4 × 4 × 4 and 4 × (L^2 Q/4), respectively. More specifically, we also compared the execution time of both secret message embedding and extraction. The results are shown in Table 10. It is apparent that most of schemes can implement experiments in <1 s, which implies that they are suitable to be used in real-time applications. It is also seen that our approach achieved a good performance in the lightweight computation, with an overall execution time of 0.591 s.

### Table 9: Comparisons of computational cost

| Images   | Computational complexity | Size of reference matrix | Size of LUT |
|----------|--------------------------|--------------------------|-------------|
| Liu et al. [14] | $O(8^2 + H^2 W)$ | 256 × 256 | 4 × 4 |
| Li et al. [18] | $O(32^2 + H^2 W)$ | 256 × 256 | 4 × 16 |
| Chen et al. [28] | $O(32^2 + H^2 W)$ | 256 × 256 | 4 × 4 × 4 |
| Ours | $O(H/W) + L^2 Q/2)$ | 256 × 256 | 4 × (L^2 Q/4) |

### Table 10: Comparisons of execution times (s)

| Images   | $T_{emb}$ | $T_{ext}$ | $T_{emb}$ | $T_{ext}$ | $T_{emb}$ | $T_{ext}$ | $T_{emb}$ | $T_{ext}$ |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Baboon   | 0.963     | 0.088     | 1.116     | 0.165     | 0.264     | 0.050     | 0.463     | 0.154     |
| Boat     | 0.895     | 0.056     | 0.984     | 0.116     | 0.187     | 0.037     | 0.461     | 0.146     |
| Elaine   | 0.766     | 0.062     | 0.900     | 0.085     | 0.186     | 0.046     | 0.440     | 0.146     |
| Goldhill | 0.802     | 0.076     | 0.992     | 0.112     | 0.160     | 0.043     | 0.437     | 0.145     |
| Lake     | 0.748     | 0.074     | 0.955     | 0.091     | 0.170     | 0.046     | 0.438     | 0.149     |
| Lena     | 0.925     | 0.067     | 0.912     | 0.086     | 0.168     | 0.040     | 0.432     | 0.146     |
| Peppers  | 0.673     | 0.050     | 0.915     | 0.085     | 0.155     | 0.039     | 0.445     | 0.145     |
| Airplane | 0.585     | 0.048     | 0.956     | 0.090     | 0.144     | 0.038     | 0.435     | 0.145     |
| Average  | 0.795     | 0.065     | 0.966     | 0.104     | 0.179     | 0.042     | 0.444     | 0.147     |
| Overall  | 0.860     | 1.070     | 0.222     | 0.591     |

$T_{emb}$: The execution time of secret messages embedding.

$T_{ext}$: The execution time of secret messages extraction.

matrix is $O(H/L_Q)(W/L_Q) + O(L^2 Q)$; the computational complexity in constructing LUT lies in the size of the spliced $Q$, which takes $O(4^2 + L^2 Q/4)$; the computational complexity in dividing pixel pairs is $O(H/W^2)$; the computational complexity in embedding or extracting secret messages is $O(H/W^2)^{(4-L^2 Q/4)}$.

Considering the $L_Q$ is usually far smaller than $H$, thus, the overall computational complexity of our approach can be represented by $O(H^2/W^2) = O(4-L^2 Q/4)$ because of the required processing of embedding secret messages with the help of a 4 × 4 location table. Similarly, the computational complexity of scheme [14] is $O(H/W^2) = O(8-H^2 W)$, where the LUT is sized 4 × 16. For scheme [28], due to the three-layer mini-Sudoku designed to serve for secret messages embedding or extraction, its computational complexity is as $O(H/W^2) = O(32-H^2 W)$. Additionally, it is also worth noting that the temporary storage cost of reference matrix for those four schemes are all sized of 256 × 256, whereas the storage cost of the LUT is sized of 4 × 4, 4 × 16, 4 × 4 × 4 and 4 × (L^2 Q/4), respectively. More specifically, we also compared the execution time of both secret message embedding and extraction. The results are shown in Table 10. It is apparent that most of schemes can implement experiments in <1 s, which implies that they are suitable to be used in real-time applications. It is also seen that our approach achieved a good performance in the lightweight computation, with an overall execution time of 0.591 s.

4.3.4 Potential failure analysis: Based on empirical results, it is noted that the proposed scheme has excellent performance, exceeding the performance of some state-of-the-art schemes in both EC and visual quality. Nevertheless, some potential failures our approach may occur when the following extreme cases were encountered.

- The length of the secret message (i.e. $L_Q$) is too long. The longer $L_Q$ is, the larger $L_Q$ should be. Doing so that, the stego-image will be seriously distorted. It means that the balance between the EC and stego-image quality is broken.

However, in fact, there are only 19 different tetrominoes at most in our system. This may result in a failure in constructing the workable reference matrix and loop-up-table.

- Without owning the prior knowledge of the usable spliced $Q$ or the rule of constructing LUT. In this case, it is difficult to extract the secret messages with error-free.

However, the above three cases can be prevented in advance by either setting rules for determining the maximimal size of the length of secret message and $L_Q$ or pressharing the knowledge of spliced $Q$ or the rule of constructing LUT.

5 Conclusions

In this paper, we proposed a novel Tetris-based DH scheme to achieve a larger EC along with better image quality. A tetromino-based falling puzzle game was played first within an $L_Q = L_Q$ square lattice $Q$. Then, according to the decided $Q$, the reference matrix and corresponding LUT are constructed for DH. The maximum EC can be flexibly selected according to the $L_Q$. The experimental results showed that our proposed scheme had a more excellent performance than other state-of-the-art schemes in both EC and the image quality. In addition, the security analysis proved our approach is secure covert communication. In the future, we will explore the novel strategy to solve the potential failures in our system. Also, we will try to investigate more applications by using the proposed Tetris-based DH mechanism, such as watermarking, image authentication, and hiding information in encrypted images.

6 Acknowledgments

This work was supported in part by the Natural Science Foundation of Fujian Province under grant no. 2018J01788, and in part by the Open Fund of Engineering Research Center for ICH Digitalization and Multi-source Information Fusion of Fujian Province under grant no. FJ-IICH201901, and in part by the Education-Scientific Research Project for Middle-aged and Young of Fujian Province under grant nos. JT180621 and JAT190488.

7 References

[1] Chang, C.C., Li, C.T., Chen, K.: ‘Privacy-preserving reversible information hiding based on arithmetic of quadratic residues’, IEEE Access, 2019, 7, pp. 54117–54132
[2] Chang, C.C., Li, C.T., Shi, Y.Q.: ‘Privacy-aware reversible watermarking in cloud computing environments’, IEEE Access, 2018, 6, pp. 70720–70733

[3] Chang, C.C., Li, C.T.: ‘Algebraic secret sharing using privacy homomorphisms for IoT-based healthcare systems’, Math. Biosci. Eng., 2019, 16, (5), pp. 3367–3381

[4] Su, G.D., Chang, C.C., Lin, C.C.: ‘A high capacity reversible data hiding in encrypted AMBTC-compressed images’, IEEE Access, 2020, 8, pp. 26984–27000

[5] Phadikar, A., Maity, S.P.: ‘ROI based error concealment of compressed object based image using QM data hiding and wavelet transform’, IEEE Trans. Consum. Electron., 2010, 56, (2), pp. 971–979

[6] Ntalians, K.S., Doulamis, A.D., Doulamis, N.D., et al.: ‘A robust steganographic wavelet-based system for resistant message hiding under error prone networks’, Proc. IEEE Int. Conf. on Multimed. & Expo, IEEE, Lausanne Switzerland, August 2002, pp. 561–564

[7] Wang, H., Tsaiharri, S.A., Katsaggelos, A.K.: ‘Joint source-channel coding for wireless object-based video communications utilizing data hiding’, IEEE Trans. Image Process., 2006, 15, (8), pp. 2158–2169

[8] Chan, C.K., Cheng, L.M.: ‘Hiding data in images by simple LSB substitution’, Pattern Recognit., 2004, 37, (3), pp. 469–474

[9] Mielikainen, J.: ‘LSB matching revisited’, IEEE Signal Process. Lett., 2006, 13, (5), pp. 285–287

[10] Zhang, X., Wang, S.: ‘Efficient steganographic embedding by exploiting modification direction’, IEEE Commun. Lett., 2006, 10, (11), pp. 781–783

[11] Kim, H.J., Kim, C., Choi, Y., et al.: ‘Improved modification direction method’, Comput. Math. Appl., 2010, 60, (2), pp. 319–325

[12] Kieu, T.D., Chang, C.C.: ‘A steganographic scheme by fully exploiting modification direction’, Expert Syst. Appl., 2011, 38, (8), pp. 10648–10657

[13] Chang, C.C., Liu, Y., Nguyen, T.S.: ‘A novel turtle shell based scheme for data hiding’, Proc. of 2014 Tenth Int. Conf. Intelligent Inf. Hiding Multimed. Signal Process., Kitakyushu Japan, August 2014, pp. 84–93

[14] Liu, Y., Chang, C.C., Nguyen, T.S.: ‘High capacity turtle shell-based data hiding’, IET Image Process., 2016, 10, (2), pp. 130–137

[15] Jin, Q., Li, Z., Chang, C.C., et al.: ‘Data hiding based on extended turtle shell matrix construction method using particle swarm optimization’, Int. J. Netw. Secur., 2017, 19, (1), pp. 154–162

[16] Liu, L., Wang, L., Chang, C.C., et al.: ‘Data hiding based on extended turtle shell matrix construction method using particle swarm optimization’, Int. J. Netw. Secur., 2017, 19, (1), pp. 12233–12250

[17] Liu, L., Wang, L., Chang, C.C.: ‘Data embedding scheme based on multi-matrix construction of turtle shell to avoid human eye perception’, Multimed. Tools Appl., 2017, 76, (10), pp. 12233–12250

[18] Li, X.S., Chang, C.C., He, M.X., et al.: ‘Secure high capacity data hiding scheme based on reference matrix’, Int. J. Netw. Secur., 2019, 21, (6), pp. 918–929

[19] Lee, C.F., Shen, J.J., Agrawal, S., et al.: ‘Data hiding method based on 3D magic cube’, IEEE Access, 2020, 8, pp. 39430–39453

[20] Chang, C.C., Chou, Y.C., Kieu, T.D.: ‘An information hiding scheme using Sudoku’, Proc. of 3rd Int. Conf. Innovative Comput. Inf. Control, Dalian China, August 2008, pp. 17–22

[21] Hong, W., Chen, T.S., Shiu, C.W.: ‘Steeganographic using Sudoku revisited’, Proc. of 2nd Int. Symp. Intelligent Inf. Tech. Appl., Shanghai China, December 2008, pp. 935–939

[22] Hong, W., Chen, T.S., Shiu, C.W.: ‘A minimal Euclidean distance searching technique for Sudoku steganography’, Proc. of 2008 Int. Symp. Inf. Sci. Eng., Shanghai China, December 2008, 1, pp. 515–518

[23] Chang, C.C., Su, G.D., Lin, C.C.: ‘An improved sudoku-based data hiding scheme using greedy method’, Int. J. Comput. Sci. Eng., 2020, 23, (1), pp. 10–21

[24] Farn, E.J., Chen, C.C.: ‘Jigsaw puzzle images for steganography’, Opt. Eng., 2009, 48, (7), p. 077006

[25] Ou, Z.H., Chen, L.H.: ‘A steganographic method based on Tetris games’, Inf. Sci., 2014, 276, pp. 343–353

[26] Lyu, Y., Yin, D., Zhang, Y., et al.: ‘A reliable data hiding scheme using jigsaw Sudoku’, Proc. of 2018 Int. Conf. Network, Commun., Comput. Eng. (NCCE 2018), Chong Qing China, May 2018, vol. 147, pp. 805–812

[27] He, M., Liu, Y., Chang, C.C., et al.: ‘A Mini-Sudoku matrix-based data embedding scheme with high payload’, IEEE Access, 2019, 7, pp. 141414–141425

[28] Chen, W., Chang, C.C., Wang, S., et al.: ‘Multi-layer mini-sudoku based high-capacity data hiding method’, IEEE Access, 2020, 8, pp. 69256–69267

[29] Horng, J.H., Xu, S., Chang, C.C., et al.: ‘An efficient data-hiding scheme based on multidimensional mini-sudoku’, Sensors, 2020, 20, (9), p. 2739

[30] Demaine, E.D., Hohenberger, S., Liben-Nowell, D.: ‘Tetris is hard, even to approximate’. Proc. of Int. Comput. Combinatorics Conf., Berlin Heidelberg, June 2003, pp. 351–363

[31] Xie, X.Z., Chang, C.C., Lin, C.C.: ‘Reversibility-oriented secret image sharing mechanism with steganography and authentication based on code division multiplexing’, IET Image Process., 2019, 13, (9), pp. 1411–1420

[32] Alias Sathya, S.P., Ramakrishnan, S.: ‘Non-redundant frame identification and keyframe selection in DWT-PCA domain for authentication of video’, IET Image Process., 2020, 14, (2), pp. 366–375

[33] Su, G.D., Chang, C.C., Lin, C.C.: ‘Effective self-recovery and tampering localization fragile watermarking for medical images’, IEEE Access, 2020, 8, pp. 160840–160857

[34] Zhang, X., Wang, S.: ‘Vulnerability of pixel-value differing steganography to histogram analysis and modification for enhanced security’, Pattern Recognit. Lett., 2004, 25, (3), pp. 331–339

[35] Fridrich, J., Goljan, M., Du, R.: ‘Detecting LSB steganography in color, and gray-scale images’, IEEE Multimed., 2001, 8, (4), pp. 22–28

[36] Su, G.D., Liu, Y., Chang, C.C.: ‘A square lattice oriented reversible information hiding scheme with reversibility and adaptivity for dual images’, J. Vis. Commun. Image Represent., 2019, 64, p. 102618

[37] Leng, H.S.: ‘Generalized scheme based on octagon-shaped shell for data hiding in steganographic applications’, Symmetry. (Basel), 2019, 11, (6), p. 760

[38] Xie, X.Z., Liu, Y., Chang, C.C.: ‘Extended squared magic matrix for embedding secret information with large payload’, Multimed. Tools Appl., 2019, 78, (14), pp. 19045–19059

[39] Lee, C.F., Li, Y.C., Chu, S.C., et al.: ‘Data hiding scheme based on a flower-shaped reference matrix’, J. Netw. Intell., 2018, 2, (2), pp. 138–151

IET Image Process., 2020, Vol. 14 Iss. 17, pp. 4633-4645
This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)