Journal of King Saud University – Computer and Information Sciences 34 (2022) 9273–9283

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Improved exploiting modification direction steganography for hexagonal image processing

Nazife Cevik [a], Taner Cevik [b,*], Onur Osman [b], Ahmet Gurhanli [b], Sajjad Nematzadeh [b], Fatih Sahin [b]

[a] Computer Engineering Dept., Faculty of Engineering and Architecture, Istanbul Arel University, Istanbul, Turkey
[b] Computer Engineering Dept., Faculty of Engineering and Architecture, Istanbul Nisantasi University, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 27 July 2022
Revised 29 August 2022
Accepted 9 September 2022
Available online 15 September 2022

Keywords:
Steganography
Hexagonal image processing
Hexel
Exploited modification direction

ABSTRACT

Steganography has made significant advances in the **S**quare-pixel-based Image Processing (SIP) domain, but to our knowledge, no work has yet been done in Hexel (Hexagonal Pixel)-based Image Processing (HIP). This paper presents a HIP-domain data hiding method that exploits and improves the SIP-domain Exploiting Modification Direction (EMD) embedding scheme. The proposed method, Hexagonal EMD (HexEMD), utilizes a HIP-domain cover image's hexagonal nature and infrastructure to embed the secret message. In standard digital imaging systems, the sensor portion that converts photonic energy into an analog electrical signal and all the subunits that digitize, process, and display this signal are based on square pixel logic, so there is currently no commercial equipment available to produce HIP-domain images. Thus, the image is first transformed into the HIP domain in software using the infrastructure developed in the project. Then the HIP-domain image is partitioned into non-overlapping heptads of the standard size, each containing seven hexels. Rather than embedding segments to the independent pixel pairs as done in SIP-domain EMD, we do the embedding iteratively in each heptad. Experimental results show that the HexEMD outperforms its SIP equivalent, EMD, by improving embedding capacity and achieving low visual quality distortion.

1. Introduction

The Internet has facilitated large-scale data transmission in recent years. Naturally, this raised concern over data security (Younus and Hussain, 2022; Wang et al., 2010). Communication security is ensured using data hiding, also called steganography and encryption (Rustad et al., 2022). Although the primary purpose of both approaches is to preserve data, there are differences in how the data is concealed and how the hidden data appears. A ciphertext is created in cryptography by encoding plain text with a specific key and then encrypting it using a particular method. The format and meaning of the communication are altered by the ciphertext, making it impossible for those who do not own the secret key to decipher the original message (Ardiansyah et al., 2017; Suresh and Sam, 2020). However, this is not the case for steganography. Steganography is the process of hiding data by incorporating information into an environment while preserving the medium's integrity to avoid external detectability (Al-Dmour and Al-Ani, 2016). Steganography has broad application areas such as military, commercial, and anti-criminal to ensure copyright protection, content authentication, and secure communication (Hostalot and Megias, 2013; Chang et al., 2008).

The term "steganography" was originated by fusing the Greek words "stego", which means "secret", and "grafia", which means "writing" (Roy et al., 2013). Among many other forms, including audio, video, and text, image is the media type most focused on as cover (Cheddad et al., 2010; Hussain et al., 2018; Setiadi, 2022). The secret data or the material to be concealed, the cover image, which will carry the data to be retained, and the stego image, which includes the resultant and embedded data, are the three primary components of the steganography process. To be deemed high-quality, a steganography technique must have low detectability of hidden data in the stego image and little distortion

* Corresponding author.
E-mail addresses: nazifecevik@arel.edu.tr (N. Cevik), taner.cevik@nisantasi.edu.tr (T. Cevik), onur.osman@nisantasi.edu.tr (O. Osman), ahmet.gurhanli@nisantasi.edu.tr (A. Gurhanli), sajjad.nematzadeh@nisantasi.edu.tr (S. Nematzadeh), fatih.sahin@nisantasi.edu.tr (F. Sahin).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2022.09.007
1319-1578/

from embedding in the cover image. (Filler et al., 2010; Lyu and Farid, 2006).

Image processing is the process of translating human vision to computer vision. The data in the light physical medium is continuous, and specific sensors are used to collect this continuous data. These sensors are employed in square or rectangular arrays and have a wide range of light sensitivity. Computers can only process digital data, despite the continuous nature of light data. Continuous light data must thus be sampled and digitalized. The usage of square or rectangular sensor arrays affects downstream computer processing. The smallest digital data unit in a computer system, the pixel, is therefore made into a square.

The biggest challenges of HIP are the storage of the obtained data, the absence of the necessary data structures to process the data, the algebra to be used as a basis, and the software components necessary to implement all these (Popescu and Tanasie, 2012; Sahr, 2011). On the other hand, changing many things and producing good outcomes may be achieved by sampling light data on a hexagonal lattice and processing it as a hexagon domain. For many years, hexagonal geometry has been researched. Before Hales (Hales, 2000; Hales, 2001) showed that they are, hexagons were not the most excellent method to partition a plane into equal-sized areas. Honeycombs are another naturally occurring hexagonal interaction with hexagonal geometry, in addition to the naturally hexagonal arrangement of photoreceptors in the fovea (Coleman et al., 2009). The hexagonal lattice architecture provides some benefits over its square brother. The employment of circular symmetric kernels, which boost detection precision for straight and curved edges, and the homogeneity of the hexagonal lattice structure, which offers local equality and uniqueness, are made possible by better radial symmetry (Allen, 2005).

1.1. Contribution of this study

Due to a lack of necessary hardware, algebraic, and software components to handle hexels, the HIP has received little attention. On the other hand, HIP has to be carefully examined to determine if it may help with the problem of data size and, therefore, processing time. HIP is also intriguing to work with since it may enhance the output quality of standard SIP techniques like edge detection, segmentation, and object identification. Despite steganography's lengthy history in the SIP domain, there has been no effort to apply or further develop it in the HIP area.

The contributions of this study can be expressed as follows:

- To our knowledge, no steganography strategy has been proposed for the HIP domain. This is the first research of its type in the field. This study presents a HIP-domain data hiding method that exploits and improves the SIP-domain Exploiting Modification Direction (EMD) embedding scheme.
- The amount of data hidden in the content and the degree of variation between the new material produced as a result of this data hiding and the original version are the two most important factors that affect how well a data hiding strategy works. According to the simulation results, steganography done in the HIP is preferable to steganography done in the SIP in terms of both the extra quantity of data that can be concealed and the less significant change in the material following data hiding.

1.2. Article outline

The remainder of the text is structured as follows. In Section 2, steganography basics and associated research are introduced. The standard SIP-domain EMD's guiding concepts are reviewed in Section 3. The suggested HIP-domain data-hiding strategy is presented in Section 4. The testing dataset, the experimental design, and explanations of the outcomes are introduced in Section 5. Section 6 concludes by outlining our findings and potential avenues for further investigation.

2. Preliminaries

Information is concealed inside other information using the art and science of steganography, a kind of covert communication. Steganographic transmissions are undetectable compared to encryption since the data is obscured to the human eye (Douglas et al., 2018). The World Wide Web's growth has considerably boosted the use of digital images. Due to many superfluous bits in the digital image representation, images are more frequently used to incorporate steganographic data. Within the realm of digital images, several distinct image file formats exist. Eventually, many steganography methods are available for every image format (Morkel et al., 2005).

In order to prevent deformation of the cover image, the most modern steganography techniques attempt to inject hidden information into the cover image's edge or texture region. The cover image will invariably show signs of modification if secret information is included. Even if the concealed message is included in the redundant texture region of the carrier image, it is not easy to completely prevent the detection of statistical modeling. Standard steganography methods, therefore, present a covert security risk. Researchers (Subhedar and Mankar, 2014) created the idea of implicit information masking to evade steganalysis detection effectively. The underlying idea is that the cover image is created using the hidden message.

Designing a novel data concealing system that achieves good visual quality, hiding capacity, resilience, and steganographic security is technically challenging. As a result, steganography has been thoroughly researched, and several techniques have been put out in the literature. Four categories are used to group image steganography techniques: spread spectrum, transform domain, spatial domain, and model-based steganography (Zhang et al., 2022; Ker, Jun. 2005).

2.1. Spatial domain steganography

The pixel value contains the concealed message right there in the spatial realm. Data concealment techniques like the least significant bit replacement method (LSB) are widely used. Due to its cheap CPU cost and simplicity of implementation, this approach is one of the most used embedding techniques. However, LSB embedding increases the intensity values of even-intensity pixels by one or leaves them unchanged, while decreasing the intensity values of odd-intensity pixels by one or leaving them unchanged. Due to this uneven embedding distortion, steganalysis can be used to detect it (Fridrich et al., 2001; Harmsen and Pearlman, 2003). It is asymmetric to use the LSB replacement approach. This asymmetry is taken advantage of in steganalysis. LSB-R can reportedly be found by some detectors (Chan and Cheng, 2004; Sharp, 2137). In 2004, Chan et al. (Ker, 2004) proposed the straightforward and effective optimal pixel adjustment procedure (OPAP) technique to remove the distortion caused by LSB replacement. A simple evaluation adjusts the other bits under their nature if message bits are stored in the rightmost LSBs of a pixel. In other words, these bits are either replaced with the adjusted result or kept unaltered if the rectified result has low distortion.

Sharp came up with the LSB matching strategy to get around the asymmetry of the LSB substitution scheme (Ker, 2005). The LSB Matching Method (LSB-M) does not only replace the LSB of an overlay pixel with a hidden bit. Instead, the coverage pixel is randomly raised or decreased by one if the secret bit does not

match the least significant bit of the coverage pixel. Pixels that are odd or even are no longer asymmetric. Using a statistical detector to identify LSB-M is far more difficult than LSB-R detection, as is well known (Kieu and Chang, 2011).

On the other hand, the LSB-M method is recognized by Ker's proposed detector (Mielikainen, 2006). By doing away with the LSB-R method's asymmetry, the LSB-M technique may achieve the same visual quality and concealing effectiveness (Zhang and Wang, Nov. 2006). To enhance the LSB-M approach's visual quality, Mielikainen presented the LSB matching revisited data embedding system (Hong and Chen, 2012). The LSB matching revisited (LSB-MR) approach uses the binary function and four embedding criteria to embed two hidden bits into a pair of cover pixels simultaneously. The payload of Mielikainen's approach is identical to that of the LSB-M method, but it needs fewer changes to the cover image. The LSB-M technique performs better than Mielikainen's system, which has an estimated number of alterations per pixel of 0.375 compared to 0.5. Therefore, as measured by the peak signal-to-noise ratio, the LSB-MR technique has better visual quality than the LSB-M method (PSNR).

A message digit in a 5-ary notational system may now be encoded in just one pixel of a pixel pair, according to Zhang and Wang's exploiting modification direction (EMD) technique (Wu and Tsai, 2003), which enhances Mielikainen's method. LSB matching and EMD algorithms may significantly enhance the traditional LSB technique, producing more outstanding stego image quality for the same payload. On the other hand, the maximum payloads for EMD and LSB matching are merely 1 and 1.161 bpp, respectively. Therefore, these two strategies are useless for applications requiring a high payload. The LSB matching and EMD embedding methods (Tseng and Leng, 2013) cannot be used to enhance the payload.

The Pixel Value Differencing (PVD) (Shen and Huang, 2015) approach provides great imperceptibility for steganographic images by calculating the charge based on the difference between successive pixels by choosing two and building a quantization range table. Additionally, it offers the benefit of transporting several payloads while preserving constant image properties after data embedding. Several recent studies have been suggested to enhance PVD (Pradhan et al., 2017; Hussain et al., 2017; Kalaivanan et al., 2015; Patil and Bormane, 2013).

2.2. Frequency domain steganography

There is a variety of low- and high-frequency components in every digital image. High-frequency content is represented by the edges and abrupt transitions, whereas the level and smooth parts represent low-frequency content. Since changes in low-frequency zones are invisible to the human eye, they are more sensitive to Human Visual System (HVS). As a result, it is difficult to conceal an equivalent amount of information in both high-frequency and low-frequency zones. Additionally, although pixels in the high-frequency region significantly deviate from their neighbors, those in the low-frequency zone are closely linked to their neighbors. We may thus conclude that obtaining and studying the image in the frequency domain will significantly increase the effectiveness of data embedding. Systems in the transform domain are less susceptible to attacks.

The discrete wavelet transform (DWT) (Mazumder and Hemachandran, 2013; Amin et al., 2014; Mitra et al., 2015), wavelet packet, and discrete cosines transform (DCT) (Attaa and Ghanbari, 2018; Abdullah et al., 2014) are some of the most frequent transforms that are used to convert the cover image into frequency domain coefficients in frequency domain approaches. The secret message is concealed among themselves by manipulating these transform coefficients. The inverse transformation is then used to produce the stego-image (Lin, 2014). The cover image is mapped to an unidentified frequency domain using a unique method described in (Seyedi et al., 2011). The obtained coefficients then contain the data encoded. Cover images in some studies are mapped using DCT. One significant issue arises when the stego-images are kept in a limited range of integers, such as 0 to 255, because the usual DCT coefficients are in the real domain. The mapping in (Houssein et al., 2016) guarantees an integer-to-integer translation in the DCT domain.

Data hiding using DWT has lately gained notoriety. DWT detail coefficients have been used for data embedding in several research studies (Singh and Siddiqui, 2012). In these studies, the data is usually sent through edge coefficients. A method based on the Haar wavelet transform is provided in (Miri and Faez, 2018), which raises the security level of the algorithm by encrypting the secret message beforehand. Then, secret message bits are replaced with the first LSB bit of coefficients. The second bits of coefficients can, if necessary, carry the remaining bits. This ensures that the content is spread evenly throughout the cover image. Authors used a different wavelet family in specific techniques, including (Marvel et al., 1999), like the Redundant Discrete Wavelet Transform (RDWT), which does not require a downsample block for computing coefficients. As a result, some signal features regarded as valuable in signal processing applications can be preserved (Sallee, 2003).

2.3. Spread spectrum steganography

Spread spectrum steganography harvests and uses techniques from various domains, including spread-spectrum communication, image restoration, error-control coding, and others. Here, embedding sensitive data into a sort of noise unique to image capture is the primary technique. A digital cover picture is then enhanced with this noise (xxxx).

2.4. Model-based steganography

The statistical model of the cover image serves as the foundation for model-based steganography. The general statistical properties of the image are first extracted. Using these statistical facts, the secret data is inserted in the proper locations (Younus and Younus, 2020).

3. Review of EMD

Zhang and Wang (Wu and Tsai, 2003) proposed EMD, one of the primary spatial domain data-hiding methods that has inspired many followers. Each secret digit in a $(2n+1)$-ary notational system is held by n cover pixels, where n is a system parameter, and, at most, only one pixel among these n pixels is decreased or increased by 1. This is the fundamental notion of EMD. Only one pixel from this set of $2n$ pixels will change as its value increases or decreases, so there are two possibilities. Thus, a total of $2n$ different results can occur. The secret message is first converted into a string of digits using an odd-base $(2n+1)$ notation method. If the secret message is in binary mode, it can be split into chunks of L bits (Eq. (1)). Then, the decimal value of each segment of these L bits is represented by K digits in the $(2n+1)$-ary notation system.

$$L = K.\log_2 2n + 1 \tag{1}$$

Let msg="AX", a two-character secret message. The binary counterpart of this two-character message, which we assume is encoded in ASCII/UTF-8, is $(msg)_2 = ''0100000101011000''$. For the values of $K=2$ and $n=3$ in a 7-ary notational system, then $L = \lfloor 2.\log_2 2.3 + 1 \rfloor = 4$. Hence, $(msg)_2$ is segmented into 4-bit-length of segments as $(msg)'_2 = (0100000101011000)_2$. Then for

the bit-translated secret message, its 7-ary equivalent is calculated as $(msg)'_7 = (04010511)_7$. For example, since $n = 3$, consider a group of three pixels as [137139141] to embed the first secret 7-ary digit $d = (1)_7$. Considering Eq. (2):

$$f(g_1, g_2, \cdots, g_n) = \left[\sum_{i=1}^{n} (g_i.i)\right] mod(2n+1) \tag{2}$$

$f(137\,139\,141) = (4)_7$. If *dequalsf*, no modification is needed on the intensity values of the pixel group. Otherwise, the difference between *fandd* is calculated as in Eq. (3):

$$s = (d - f) mod(2n+1) \tag{3}$$

In our example, that is $s = (1-4)mod(7) = 4$. In EMD, if $s \leq ntheng_s = g_s + 1$, otherwise $g_{2n+1-s} = g_{2n+1-s} - 1$. Thus, since $4 \leq 3$ in our example, $g_3 = g_3 - 1 = (137 - 1) = 136$. The values of the remaining pixels in the triple do not change, and the resulting stego-pixels-triple becomes [136 139 141].

As an extreme case, if the pixel is saturated, an increase in g_s or reduction in g_{2n+1-s} may not be permitted. At this point, the value of the saturated pixel should be changed by 1, and embedding is done accordingly. Assume that we have a pixel-group [255255255254], $n = 4$, and $d = (0)_9$. In this case, $f(255\,255\,255\,254) = (8)_9$ and $s = (0-8)mod(9) = 1$. Since $1 \leq 4$, $g_1 = g_1 + 1 \rightarrow 255 + 1$, which is not allowed. Thus, $g_1 = g_1 - 1 \rightarrow 255 - 1$, and the pixel group becomes [254255255254]. The process continues with the right-next pixel. For the updated pixel group, $f(254\,255\,255\,254) = (7)_9$ and $s = (0-7)mod(9) = 2$. Since $2 \leq 4$, $g_2 = g_2 + 1 \rightarrow 255 + 1$, which is not allowed again. Thus, $g_2 = g_2 - 1 \rightarrow 255 - 1$, and the pixel group becomes [254254255254]. For the updated pixel group, $f(254\,254\,255\,254) = (5)_9$ and $s = (0-5)mod(9) = 4$. Since $4 \leq 4$, $g_4 = g_4 + 1 \rightarrow 254 + 1 = 255$. This time $d = (0)_9$ is successfully embedded, and the final state of the pixel group becomes [254254255255]. While saturation may call for the change of several pixels, this is ignored since it has little impact on performance because saturated pixels in a native image are uncommon.

4. The proposed method

Due to a lack of hardware, algebra, and software infrastructure, working on HIP is a complex and challenging sector. Every subject used in conventional SIP must likewise have a HIP equivalent. One of them is steganography. As was already established, the spatial domain is where image-steganography began to be used, and here is also where the most fundamental steganographic techniques can be found. EMD is one of the most important, extensively applied, and variant-derived techniques in spatial-domain steganography. It opened the path for several subsequent studies. To our knowledge, the HIP area has not yet presented a steganography approach. This work is groundbreaking in this area.

In this section, we first talk about the set hexagonal structure and then continue with the presentation of the SIP-domain-EMD-inspired embedding scheme.

4.1. HIP infrastructure

The images in the HIP domain contain regular hexels, equivalent to the SIP definition of a pixel. Hexels' exceptional qualities make them a potential alternative method for communicating visual information. The optimum conditions for physical infrastructure include obtaining intensity and color information from a hexel-supported camera sensor and displaying them on a hexel-supported monitor. There were not many publicly accessible items when this article was written. Therefore, we projected pixels to hexels using mimic procedures. Hexagons are seen in both horizontal and vertical orientations in Fig. 1. These orientations are where the hexagonal tiling's horizontal and vertical arrangements come from.

Hexagonal patterns and their surrounding hexagons are impacted by hexagonal orientation. There are three axes with a $\pi/6$ difference in the six sides of a hexagon. Examples of tiled arrangements in horizontal and vertical orientations are shown in Fig. 2. Two oblique axes in this investigation are identified as β and γ. In addition, the α-axis indicates either the horizontal or vertical axis depending on the orientation of the hexels. The α axis is 0° for vertical hexels (Fig. 2.a) and 90° (Fig. 2.b) for horizontal hexels.

4.1.1. SIP to HIP projection

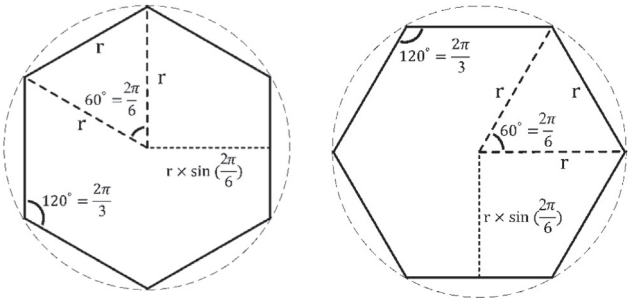
It is important to note that obtaining intensity information from a camera sensor with hexagonal support is optimal for hexagonal images. Due to the lack of such hardware, we used two distinct approaches to transform an image received as a square pixel from the SIP domain to the HIP domain. The first technique, called circular averaging, calculates the average intensity for each hexel using a circular band of pixels. Although this approach requires more processing power, it is accurate. Fig. 3 displays the representations of this method.

For even (alternate) columns, the alternate averaging approach transfers odd columns to the output matrix and calculates the average intensities of two nearby vertical pixels. It is less accurate, even if it is quicker than the previous approach. Additionally, a HIP matrix identical in size to the original SIP image is produced using this technique. The visual for this projection strategy using an alternate column average is shown in Fig. 4.

4.1.2. Coordinates and indexing

Hexagonal coordinates are not fitted in the same manner as pixel coordinates. The idea of a matrix works perfectly for pixels. However, the hexels are hampered by the absence of a functional data structure. This study suggests a memory-friendly way for indexing and storing an image's hexadecimal data. Despite the infrastructure for storing hexagonal pictures in matrices, specific calculations are necessary to identify a hexagon's neighbors. The coordinate and indexing data are displayed in Fig. 5. Utilizing the values of β and γ, one may determine the coordinate of the α -axis. Therefore, this method does not store the α value. The indexing of γ begins in the top left corner and moves down the right side, while of β begins in the upper right corner and moves leftward.

HexelToIndex and *IndexToHexel* functions transform the coordinate of a hexel to the ordinary index and the ordinary index into the hexel coordinate, respectively. In other words, these functions connect the coordinate of a given hexel to the hexel's associated



(a) Vertical hexel **(b)** Horizontal hexel

Fig. 1. Vertical and horizontal hexel orientations.

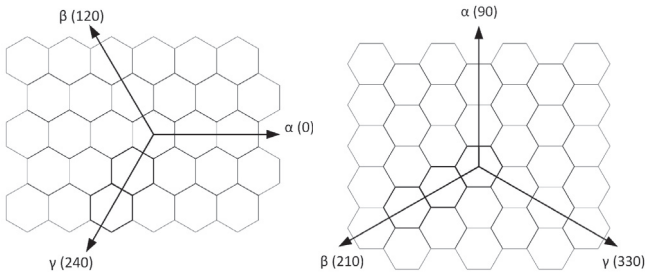


Fig. 2. Axes of hexagonal layouts in (a) vertical and (b) horizontal orientations.

index in the storage matrix. Eq. (4–5) denotes the calculations of Hexel-to-index and index-to-hexel coordinates (β, γ).

$$\gamma_T = w - \frac{(w+1) \mod 2}{2} \tag{4}$$

$$\gamma_S = \begin{cases} (\gamma - \gamma_S) * 2 if \gamma > \gamma_T \\ (\gamma_S - \gamma) * 2 else \end{cases}$$

$$step = \beta - |\gamma - \gamma_T|$$

$$index = max \begin{cases} (w+2) \times \lfloor \frac{step}{2} \rfloor + step \bmod 2 + \gamma_S \\ 2\beta - step \end{cases}$$

$$r = \lfloor \frac{index}{w} \rfloor \tag{5}$$

$$\gamma = r + \lfloor \frac{w-1}{2} \rfloor - \lfloor \frac{index}{2} \bmod \frac{w}{2} \rfloor$$

$$\beta = r + \lfloor \frac{index - r * w + 1}{2} \bmod \frac{w+1}{2} \rfloor$$

4.1.3. Traversing

Traversing is the following action. Finding a pixel's neighbors is essential for carrying out the most fundamental operations in image processing. While getting neighbors for pixels in conventional square matrices is simple, getting neighbors for hexel is more complex, especially for several levels. To quickly retrieve all of a hexel's neighbors, a particular function is created that returns all of them for a given set of tiers. Fig. 6 illustrates a hexel's traversing mechanism and three-tier neighbors.

4.2. Data embedding

The cover image is separated into groups of n pixels in standard EMD. However, as seen in Fig. 7, the cover image of our method is split into heptads, each of which consists of seven hexels.

All embedding processes are based on the reference hexel, the one in the center of each heptad after the cover image has been split into heptads. Data embedding is done on each group of n pixels in conventional PVD. Thus, for each group, only one embedding procedure exists. In our method, the leading decisive figure is the heptad. Thus, the number of embedding processes occurring on heptads differs depending on the value of n, as expressed in Table 1, where # refers to the number of embedding operations done on a single heptad.

In HexEMD, data hiding processes are made specific to each heptad. That is, depending on the value of n and K, the value of L is calculated, and accordingly, L-bits portion is extracted from the whole message. Then, this L-bits message is hidden in the hexels that make up the heptad, according to the rules in Table 1.

The procedure of the entire methodology and a single embedding process on a heptad are illustrated in Figs. 7-8 and described as follows:

Step 1: Convert the RGB cover image to grayscale.
Step 2: Transform the cover image from SIP domain to the HIP domain.
Step 3: Cluster the HIP domain image into heptads.

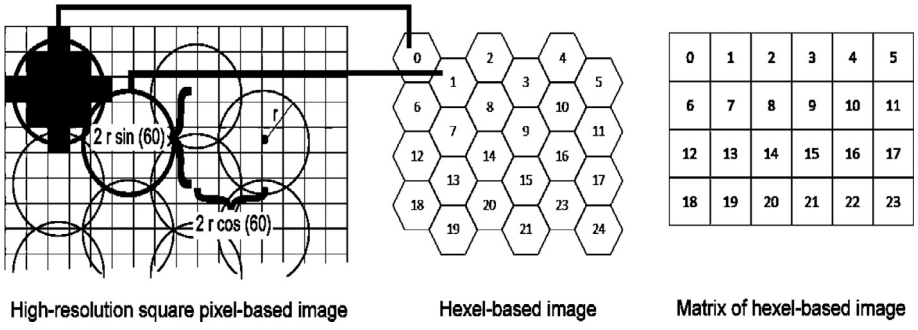


Fig. 3. Transforming a high-resolution square pixel-based image into a hexel-based image using circular occupancy.

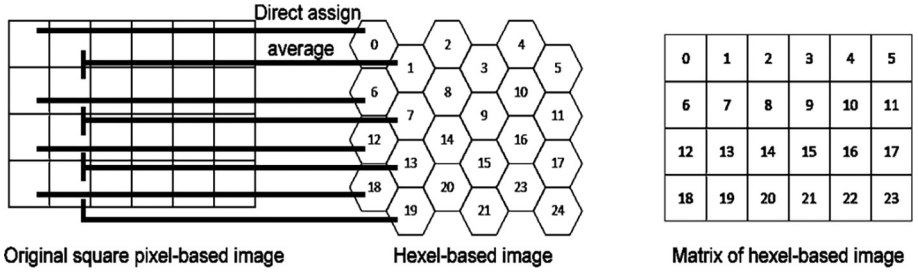


Fig. 4. Transforming a SIP image into a HIP image using the mean of alternate columns.

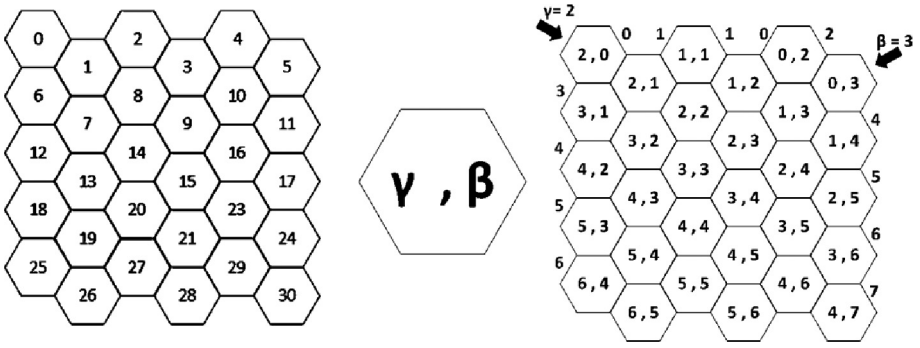


Fig. 5. Coordinates and indexing of hexels.

Step 4: Define the index values of the center hexel of each heptad.
Step 5: Pad the HIP domain cover image according to the following conditions:

- If (max(indices(:,1)) = num_rows) pad the HIP domain cover image in the row direction.
- If(max(indices(:,2)) = num_cols) pad the HIP domain cover image in the row direction.

Step 6: Calculate L considering Eq. (1).
Step 7: Calculate the total message length to be embedded. $len_{msg_{bin}} = \#_{heptad} * n * L$, where $len_{msg_{bin}}$ and $\#_{heptad}$ refer to the length of the whole binary message to be embedded in terms of bits and the number of heptads in the HIP domain cover image, respectively.
Step 8: Convert the original text message to binary form..$msg_{txt} \rightarrow msg_{bin}$
Step 9: Segment the binary converted message into L-length chunks.
Step 10: Extract L-length chunks and embed them in the heptads according to the rules listed in Table 1.

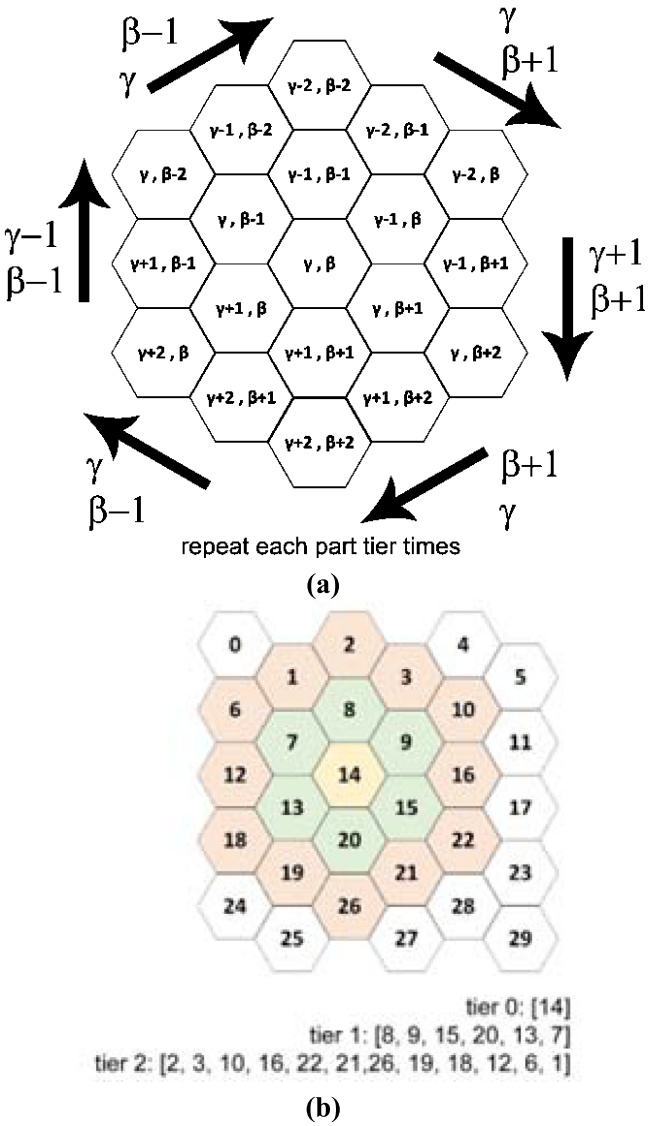


Fig. 6. (a) Traverse mechanism (b) 3-tier-neighbors of a hexel.

4.3. Data extraction

The secret digit may be readily recovered on the receiving side by computing the extraction function of the stego-pixel-group. We must determine the f value following Eq. (2–3) utilizing a collection of altered pixel values to decode the hidden message from the stego-image. However, unlike the EMD, on each heptad, data embedding is done in sequential order. That is, for $n = 2$, the embedding process is done in the order of $HexEMD(h_r, ngb_1) \rightarrow HexEMD(h_r\prime, ngb_2) \rightarrow HexEMD(h_r\prime\prime, ngb_3) \rightarrow HexEMD(h_r^{\prime\prime\prime}, ngb_4) \rightarrow HexEMD(h_r\prime\prime\prime\prime, ngb_5) \rightarrow HexEMD(h_r^{\prime\prime\prime\prime\prime}, ngb_6)$.The updated version of h_r that is $h_r\prime$ is fed to the second embedding process as input, and the output of the second process is fed to the third, which goes on like this. Therefore, the extraction process should be done in the reverse order:

$Ext\_HexEMD(h_r^{\prime\prime\prime\prime\prime}, ngb_6) \rightarrow Ext\_HexEMD(h_r\prime\prime\prime\prime, ngb_5) \rightarrow Ext\_HexEMD(h_r^{\prime\prime\prime}, ngb_4) \rightarrow Ext\_HexEMD(h_r\prime\prime, ngb_3) \rightarrow Ext\_HexEMD(h_r\prime, ngb_2) \rightarrow .Ext\_HexEMD(h_r, ngb_1)$

5. Experimental results and discussions

The main objective of this work is to simultaneously protect image quality and conceal a significant amount of information with

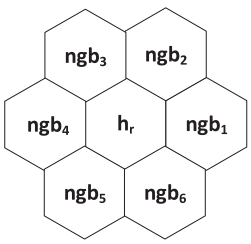


Fig. 7. An example heptad of hexels.

Table 1
Hexel-group operations performed on each heptad for varying values of *n*.

n	#	Operating group of hexels
2	6	$(h'_r, ngb'_1) = HexEMD(h_r, ngb_1)$ $(h''_r, ngb'_2) = HexEMD(h'_r, ngb_2)$ $(h''_r, ngb'_3) = HexEMD(h''_r, ngb_3)$ $(h''_r, ngb'_4) = HexEMD(h''_r, ngb_4)$ $(h''_r, ngb'_5) = HexEMD(h''_r, ngb_5)$ $(h''_r, ngb'_r) = HexEMD(h''_r, ngb_6)$
3	3	$(h'_1, ngb'_1, ngb'_2) = HexEMD(h_r, ngb_1, ngb_2)$ $(h''_r, ngb'_3, ngb'_r) = HexEMD(h'_r, ngb_3, ngb_4)$ $(h''_r, ngb'_5, ngb'_6) = HexEMD(h''_r, ngb_5, ngb_6)$
4	2	$(h'_r, ngb'_1, ngb'_2, ngb'_3) = HexEMD(h_r, ngb_1, ngb_2, ngb_3)$ $(h'_r, ngb'_4, ngb'_5, ngb'_6) = HexEMD(h_r, ngb_4, ngb_5, ngb_6)$
5	2	$(h'_r, ngb'_1, ngb'_2, ngb'_3, ngb'_4) = HexEMD(h_r, ngb_1, ngb_2, ngb_3, ngb_4)$ $(h'_r, ngb'_3, ngb'_4, ngb'_5, ngb'_6) = HexEMD(h_r, ngb'_3, ngb'_4, ngb'_5, ngb'_6)$
6	2	$(h'_r, ngb'_1, ngb'_2, ngb'_3, ngb'_4, ngb'_5) = HexEMD(h_r, ngb_1, ngb_2, ngb_3, ngb_4, ngb_5)$ $(h'_r, ngb''_2, ngb''_3, ngb''_4, ngb''_5, ngb'_6) = HexEMD(h_r, ngb'_2, ngb'_3, ngb'_4, ngb'_5, ngb_6)$
7	1	$(h'_r, ngb'_1, ngb'_2, ngb'_3, ngb'_4, ngb'_5, ngb'_6) = HexEMD(h_r, ngb_1, ngb_2, ngb_3, ngb_4, ngb_5, ngb_6)$

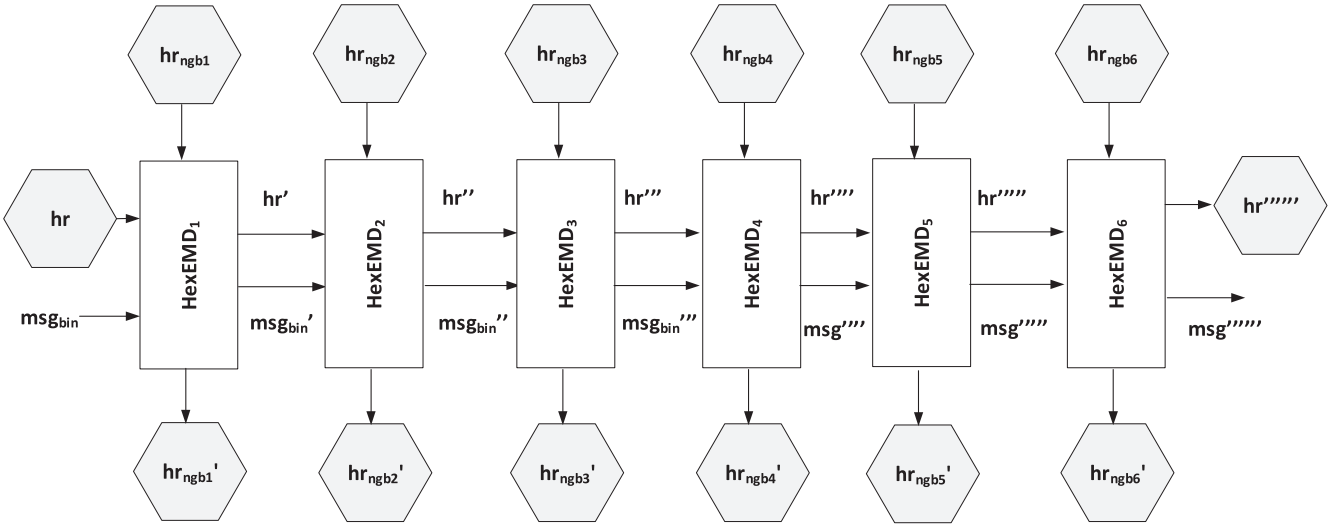


Fig. 8. The flow diagram of the HexEMD message embedding process on a single heptad.

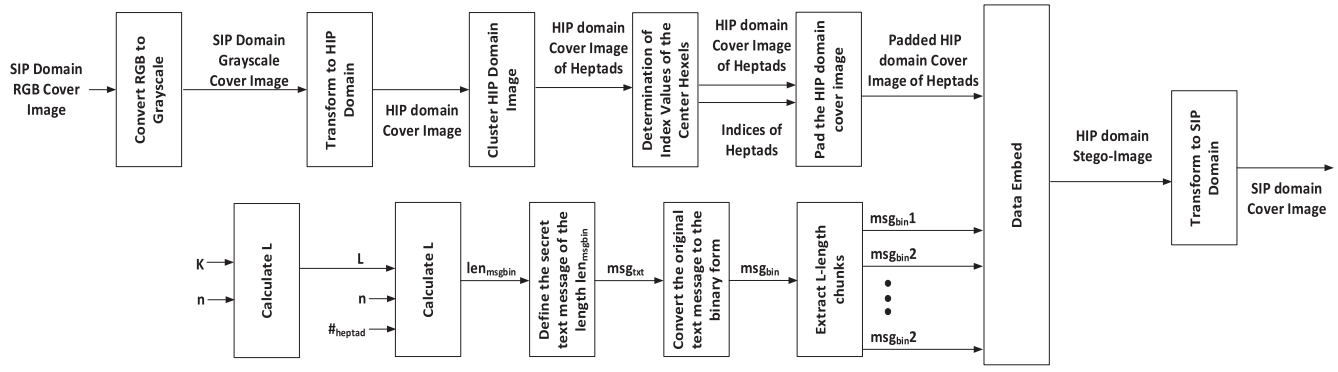


Fig. 9. The flow diagram of the entire methodology.

excellent security. Six images from the USC-SIPI Data Base (USC-SIPI) [57] were used to assess the proposed system using MATLAB 2021b as the programming language. The images utilized in this research are displayed in Fig. 9.Fig. 10..

The cover images illustrated above were resized to 256x256 pixels and then converted to grayscale. The secret message was randomly generated once, and the same message was applied to all embedding processes to ensure uniformity and fairness. Simulations and analyses were done twofold. The first is the measurement of the detectability of data hiding, and the other is the histogram analysis to see its resistance to steganalysis attacks.

Fig. 10. The images used as cover in simulations.

5.1. Imperceptibility analysis

As can be understood from its definition, the nature of data embedding in steganography is confidential. Unlike encryption, any third party should not visually notice changes to the image. Since the detection process will vary from person to person, as it is known in the literature, the most effective way to measure this is to measure the peak signal-to-noise ratio (PSNR) and mean square error (MSE) values [58].

$$MSE = \sum_{k=1}^{r\times c} \left(p_k - p_k'\right)^2 / (r \times c) \tag{6}$$

where $r \times c$ stands for the size of the image, and p_k, $p_k\prime$ are estimates of the pixels before and after data has been embedded in the image.

$$PSNR = 10\log_{10} \frac{max^2}{MSE} \tag{7}$$

Table 2 represents the *MSE* and *PSNR* values obtained for varying values of K and n.

MSE and PSNR are inversely proportional to each other. That is, PSNR decreases as MSE increases or PSNR increases as MSE decreases. Also, the capacity, that is, the amount of data embedded, is directly proportional to the MSE. That is, MSE naturally increases as the amount of hidden data increases. Because as the data is concealed, the created stego-image moves away from the original cover image. As represented in Table 2, the minimum MSE is obtained for the values of $K = 2$ and $n = 6, 7$, which is very natural because here 6–7 pixels are used to hide a hidden digit, while the other 2 pixels are only used and have a higher data hiding capacity. This issue also causes the resulting stego-image to decompose more than the original cover image. As can be seen from the table, hexEMD achieves lower MSE and higher PSNR performance compared to EMD, despite the increase in data embedding capacity, except for $K = 1$ and $n = 2$ values only. Moreover, this means achieving the ultimate goal of hiding more data and less deterioration of visual quality, thus preventing it from being discernible by third parties.

5.2. Resistance to detection analysis

When assessing the effectiveness of steganography, anti-detection performance is a crucial factor. The histogram is considerably changed when a procedure is applied to an image, making detecting a hidden message easier. We counted the probability of zeros in the difference between the cover and stego-images to more thoroughly study and contrast the detection resistance of SIP domain EMD and HIP domain HexEMD.

Naturally, a more significant percentage of zeros indicates that a small change occurred in the original image's histogram, indicating higher resistance against detection.

As identified in Table 2, HexEMD achieves higher values of the percentage of zeros, which makes the stego-image less vulnerable to histogram-based steganalysis attacks. Table 2 also indicates that for any values of K and n, the $HistSim(\%)$ values for the six cover images are always obtained in the descending order as $HistSim(\%)_{Airplane} > HistSim(\%)_{Baboon} > HistSim(\%)_{House} > HistSim(\%)_{Lena} > HistSim(\%)_{Boat} > HistSim(\%)_{Peppers}$. That is because the entropies, which denote the randomness in the images, of these cover images in ascending order are as $S_{Airplane} > S_{Baboon} > S_{House} > S_{Lena} > S_{Boat} > S_{Peppers}$. Again, HexEMD achieves a higher $HistSim(\%)$ compared to the ordinary EMD for the same parameter values, which makes HexEMD less vulnerable to stegoanalysis attacks.

5.3. Data embedding capacity analysis

As mentioned previously, the main objective of a steganography operation is to simultaneously protect image quality and hide a significant amount of information with excellent security. The data embedding capacity is measured as the number of bits hidden in

Table 2
Visual quality comparison for varying values of K and n.

K = 1, n = 4	EMD				HexEMD			
	Capacity(bits)	**MSE**	**PSNR(db)**	**HistSim(%)**	**Capacity(bits)**	**MSE**	**PSNR(db)**	**HistSim(%)**
Lena	22,016	0.1095	54.6841	89.2275	43,350	0.0974	55.1877	90.8188
Baboon	22,016	0.1099	54.6927	89.3608	43,350	0.0999	55.1911	90.9531
Airplane	22,016	0.1109	54.7133	96.1392	43,350	0.0994	55.1914	96.6751
Boat	22,016	0.1116	54.6571	88.5886	43,350	0.0982	55.1894	90.3687
House	22,016	0.1101	54.6324	95.9015	43,350	0.0986	55.1867	96.5134
Peppers	22,016	0.1126	54.6359	88.4296	43,350	0.0980	55.2018	90.1642
K = 1, n = 5	**EMD**				**HexEMD**			
	Capacity(bits)	**MSE**	**PSNR(db)**	**HistSim(%)**	**Capacity(bits)**	**MSE**	**PSNR(db)**	**HistSim(%)**
Lena	23,040	0.0904	55.5273	91.3122	43,350	0.1013	55.1931	91.1758
Baboon	23,040	0.0909	55.4830	90.8348	43,350	0.0999	55.2586	91.3422
Airplane	23,040	0.0897	55.6007	96.7998	43,350	0.1012	55.2429	96.7712
Boat	23,040	0.0897	55.5532	90.5049	43,350	0.1009	55.2039	90.8203
House	23,040	0.0920	55.5255	96.7839	43,350	0.0992	55.2398	96.6675
Peppers	23,040	0.0911	55.4159	90.4442	43,350	0.0988	55.2507	90.6403
K = 1, n = 6	**EMD**				**HexEMD**			
	Capacity(bits)	**MSE**	**PSNR(db)**	**HistSim(%)**	**Capacity(bits)**	**MSE**	**PSNR(db)**	**HistSim(%)**
Lena	22,272	0.0747	56.3364	92.6799	43,350	0.1014	55.295	91.4734
Baboon	22,272	0.0766	56.1943	92.3910	43,350	0.1033	55.2337	91.5649
Airplane	22,272	0.0765	56.2189	97.1893	43,350	0.1039	55.2565	96.8582
Boat	22,272	0.0749	56.2792	92.1486	43,350	0.1039	55.2402	59.636
House	22,272	0.0766	56.2957	97.2866	43,350	0.1015	55.3012	96.843
Peppers	22,272	0.0781	56.1582	91.8298	43,350	0.1021	55.2836	90.7745
K = 1, n = 7	**EMD**				**HexEMD**			
	Capacity(bits)	**MSE**	**PSNR(db)**	**HistSim(%)**	**Capacity(bits)**	**MSE**	**PSNR(db)**	**HistSim(%)**
Lena	23,296	0.0677	56.8732	93.3994	21,675	0.0509	58.3535	95.3629
Baboon	23,296	0.0684	56.7994	93.1770	21,675	0.0520	58.3521	95.4163
Airplane	23,296	0.0653	56.9266	97.7321	21,675	0.0505	58.3970	98.2834
Boat	23,296	0.0658	56.9294	93.3336	21,675	0.0510	58.3710	95.1782
House	23,296	0.0645	56.9549	97.7865	21,675	0.0513	58.3493	98.2483
Peppers	23,296	0.0695	56.8003	93.1648	21,675	0.0507	58.3675	94.9753
K = 2, n = 2	**EMD**				**HexEMD**			
	Capacity(bits)	**MSE**	**PSNR(db)**	**HistSim(%)**	**Capacity(bits)**	**MSE**	**PSNR(db)**	**HistSim(%)**
Lena	27,646	0.2021	52.0847	80.3696	173,400	0.2598	50.9037	82.9681
Baboon	27,646	0.2013	52.0894	80.7388	173,400	0.2699	50.8386	83.3801
Airplane	27,646	0.2017	52.0984	92.7841	173,400	0.2673	50.8936	93.7790
Boat	27,646	0.1957	52.0479	79.5837	173,400	0.2681	50.9002	82.2845
House	27,646	0.1944	52.1929	92.6849	173,400	0.2682	50.9182	93.6523
Peppers	27,646	0.1969	52.1121	78.9459	173,400	0.2655	50.8678	81.6833
K = 2, n = 3	**EMD**				**HexEMD**			
	Capacity(bits)	**MSE**	**PSNR(db)**	**HistSim(%)**	**Capacity(bits)**	**MSE**	**PSNR(db)**	**HistSim(%)**
Lena	24,660	0.1422	53.5320	86.2572	108,375	0.1434	53.5981	87.7563
Baboon	24,660	0.1446	53.5316	85.7075	108,375	0.1425	53.6131	88.0051
Airplane	24,660	0.1474	53.5178	94.9320	108,375	0.1419	53.5909	95.5750
Boat	24,660	0.1495	53.4632	85.0593	108,375	0.1411	53.6035	87.2559
House	24,660	0.1408	53.6214	95.0854	108,375	0.1395	53.6147	95.3644
Peppers	24,660	0.1432	53.6307	85.1635	108,375	0.1436	53.5596	86.8958
K = 2, n = 4	**EMD**				**HexEMD**			
	Capacity(bits)	**MSE**	**PSNR(db)**	**HistSim(%)**	**Capacity(bits)**	**MSE**	**PSNR(db)**	**HistSim(%)**
Lena	20,487	0.1095	54.6841	89.2275	86,700	0.0990	55.2235	90.8768
Baboon	20,487	0.1099	54.6927	89.3608	86,700	0.0970	55.2059	91.0233
Airplane	20,487	0.1109	54.7133	96.1392	86,700	0.0973	55.2049	96.6476
Boat	20,487	0.1116	54.6571	88.5886	86,700	0.0990	55.1998	90.4388
House	20,487	0.1101	54.6324	95.9015	86,700	0.0977	55.2279	96.5286
Peppers	20,487	0.1126	54.6359	88.4296	86,700	0.0980	55.1961	90.1291
K = 2, n = 5	**EMD**				**HexEMD**			
	Capacity(bits)	**MSE**	**PSNR(db)**	**HistSim(%)**	**Capacity(bits)**	**MSE**	**PSNR(db)**	**HistSim(%)**
Lena	21,084	0.0918	55.5331	91.1531	86,700	0.1007	55.2262	91.2460
Baboon	21,084	0.0931	55.5097	91.0619	86,700	0.1008	55.2029	91.3895
Airplane	21,084	0.0893	55.5744	96.7896	86,700	0.1003	55.2103	96.6599
Boat	21,084	0.0914	55.5446	90.5266	86,700	0.0999	55.1857	90.7898
House	21,084	0.0902	55.6172	96.7954	86,700	0.1016	55.2337	96.6614
Peppers	21,084	0.0958	55.3801	90.5049	86,700	0.0986	55.2364	90.6311
K = 2, n = 6	**EMD**				**HexEMD**			
	Capacity(bits)	**MSE**	**PSNR(db)**	**HistSim(%)**	**Capacity(bits)**	**MSE**	**PSNR(db)**	**HistSim(%)**
Lena	18,095	0.0763	56.3468	92.7341	101,150	0.1022	55.2330	91.4230
Baboon	18,905	0.0773	56.2814	92.3578	101,150	0.1032	55.2221	91.5253

(*continued on next page*)

Table 2 (*continued*)

K = 1, n = 4	EMD				HexEMD			
Airplane	18,905	0.0747	56.3181	97.3415	101,150	0.1039	55.2477	96.8353
Boat	18,905	0.0786	56.2855	92.0444	101,150	0.1021	55.2658	91.0538
House	18,905	0.0797	56.1554	97.3615	101,150	0.1031	55.2651	96.7773
Peppers	18,905	0.0760	56.2543	91.9643	101,150	0.1028	55.2340	90.8539
K = 2, n = 7	**EMD**				**HexEMD**			
	Capacity(bits)	**MSE**	**PSNR(db)**	**HistSim(%)**	**Capacity(bits)**	**MSE**	**PSNR(db)**	**HistSim(%)**
Lena	20,190	0.0634	57.0916	93.6822	50,575	0.0502	58.3850	95.3964
Baboon	20,190	0.0651	56.8971	93.0219	50,575	0.0512	58.3780	95.4300
Airplane	20,190	0.0631	57.0303	97.9114	50,575	0.0518	58.3689	98.2635
Boat	20,190	0.0667	56.8466	92.8771	50,575	0.0513	58.3417	95.1172
House	20,190	0.0679	56.8500	97.9021	50,575	0.0509	58.3647	98.2651
Peppers	20,190	0.0665	56.9276	92.8846	50,575	0.0507	58.3188	95.0150

the cover image. In EMD and HexEMD, the highest data hiding capacity is obtained for values of $K = 2$, $N = 2$. HexEMD performs much better than the EMD regarding data embedding capacity while preserving the visual quality of the stego-image.

6. Conclusion

The HIP-domain data concealing technique presented in this research takes use of and enhances the SIP-domain Exploiting Modification Direction (EMD) embedding methodology. The suggested technique embeds the hidden message using the hexagonal structure and infrastructure of a HIP-domain cover image. There is presently no commercially available equipment to make HIP-domain pictures since the sensor part of conventional digital imaging systems, as well as all the subunits that digitize, process, and display this data, are based on square pixel logic. Thus, utilizing the infrastructure created for the project, the picture is first converted into the HIP domain in software. The HIP-domain image is then divided into non-overlapping, standard-sized heptads with seven hexels in each. Unlike SIP-domain EMD, which embeds segments to separate pixel pairs, we embed segments repeatedly in each heptad. According to experimental findings, the suggested technique outperforms the SIP counterpart by increasing embedding capacity and attaining minimal visual quality distortion.

Further work is desirable to develop more HIP-compatible steganography methods by referencing this study and using the established infrastructure.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

A Pradhan, K.R. Sekhar, G. Swain, "Adaptive PVD Steganography Using Horizontal, Vertical, and Diagonal Edges in Six-Pixel Blocks", Secur. Commun. Netw., vol. 2017, Article ID 1924618, 13 pages, 2017.

Abdullah, W.M., Rahma, A.M.S., Pathan, A.S.K., 2014. Mix column transform based on irreducible polynomial mathematics for color image steganography: a novel approach. Comput. Electr. Eng. 40 (4), 1390–1404.

Al-Dmour, H., Al-Ani, A., 2016. A steganography embedding method based on edge identification and XOR coding. Expert Syst. Appl. 46, 293–306. https://doi.org/10.1016/j.eswa.2015.10.024.

J.D. Allen, 2005. "Perfect Reconstruction Filter Banks for the Hexagon Grid", In Proceedings of the 5th International Conference on Information Communications & Signal Processing, Bangkok, Thailand.

Amin, M., Abdullkader, H.M., Ibrahem, H.M., Sakr, A.S., 2014. A steganographic method based on DCT and new quantization technique. Int. J. Netw. Secur. 16 (4), 265–270.

Ardiansyah, G., Sari, C.A., Setiadi, D.R.I.M., Rachmawanto, E.H., 2017. Hybrid method using 3-DES, DWT and LSB for secure image steganography algorithm. In: In Proceedings of the IEEE 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), pp. 249–254. https://doi.org/10.1109/ICITISEE.2017.8285505.

Attaa, R., Ghanbari, M., 2018. A high payload steganography mechanism based on wavelet packet transformation and neutrosophic set. J. Vis. Commun. Image Represent. 53, 42–54.

Chan, C.K., Cheng, L.M., 2004. Hiding data in images by simple LSB substitution. Pattern Recogn. 37 (3), 469–474.

Chang, K.C., Chang, C.P., Huang, P.S., Tu, T.M., 2008. A Novel Image Steganographic Method Using Tri-way Pixel-Value Differencing. Journal of Multimedia 3, 37–44.

Cheddad, A., Condell, J., Curran, K., Kevitt, P.M., 2010. Digital image steganography: survey and analysis of current methods. Signal Process. 90 (3), 727–752. https://doi.org/10.1016/j.sigpro.2009.08.010.

Coleman, S., Scotney, B., Gardiner, B., 2009. Processing Hexagonal Images in a Virtual Environment vol 5716. https://doi.org/10.1007/978-3-642-04146-4_98.

Douglas, M., Bailey, K., Leeney, M., Curran, K., 2018. An overview of steganography techniques applied to the protection of biometric data. Multimed. Tool. Appl. 77, 17333–17373.

T. Filler, J. Judas, J. Fridrich, "Minimizing embedding impact in steganography using trellis-coded quantization", Proceedings of SPIE, Media Forensics and Security, vol. 7541, 2010.

Fridrich, J., Goljan, M., Du, R., 2001. Detecting steganography in color and grayscale images. IEEE Multimedia 8 (4), 22–28.

Hales, T.C., 2000. Cannonballs and honeycombs. Not. Am. Math. Soc. 47 (4), 440–449.

Hales, T.C., 2001. The honeycomb conjecture. Discrete Comput. Geometry 25, 1–22.

Harmsen, J., Pearlman, W., 2003. Steganalysis of additive-noise modelable information hiding. In Proceedings of SPIE: Security and watermarking of multimedia contents 5020, 131–142.

Hong, W., Chen, T.S., 2012. A Novel Data Embedding Method Using Adaptive Pixel Pair Matching. IEEE Trans. Inf. Forensics Secur. 7 (1), 176–184.

Hostalot, D.L., Megias, D., 2013. LSB Matching Steganalysis based on Patterns of Pixel Differences and Random Embedding. Comput. Secur. 32, 192–206.

Houssein, E.H., Ali, M.A., Hassanien, A.E., 2016. An image steganography algorithm using Haar discrete wavelet transform with advanced encryption system. In: Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 641–644.

Hussain, M., Wahab, A.W.A., Ho, A.T.S., Javed, N., Jung, K.H., 2017. A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement. Image Commun. 50C, 44–57. https://doi.org/10.1016/j.image.2016.10.005.

Hussain, M., Wahab, A.W.A., Idris, Y.I.B., Ho, A.T.S., Jung, K.-H., 2018. Image steganography in spatial domain: a survey. Signal Process. Image Commun. 65, 46–66. https://doi.org/10.1016/j.image.2018.03.012.

Kalaivanan, S., Ananth, V., Manikandan, T., 2015. A survey on digital image steganography. Int. J. Emerg. Trend. Technol. Comput. Sci. 4 (1), 30–33.

A.D. Ker, 2004. "Improved detection of LSB steganography in grayscale images", In Proceedings of 6th international workshop on information hiding, LNCS, vol. 3200, pp. 97–115, Springer.

Ker, A.D., 2005. Steganalysis of LSB matching in grayscale images. IEEE Signal Process Lett. 12 (6), 441–444.

Ker, A.D., Jun. 2005. Steganalysis of LSB matching in grayscale images. IEEE Signal Process Lett. 12 (6), 441–444.

Kieu, T.D., Chang, C.C., 2011. A steganographic scheme by fully exploiting modification directions. Expert Syst. Appl. 38 (8), 10648–10657.

Lin, Y.K., 2014. A data hiding scheme based upon DCT coefficient modification. Comput. Stand Interfaces 36 (5), 855–862.

Lyu, S., Farid, H., 2006. Steganalysis using high-order image statistics. IEEE Trans. Inf. Forensics Secur. 1, 111–119.

Marvel, L.M., Boncelet Jr., C.G., Retter, C.T., 1999. Spread Spectrum Image Steganography. IEEE Trans. Image Process. 8 (8), 1075–1083.

Mazumder, J., Hemachandran, K., 2013. A high capacity and secured color image steganographic technique using discrete wavelet transformation. Int. J. Comput. Sci. Informat. Technol. 4 (4), 583–589.

Mielikainen, J., 2006. LSB matching revisited. IEEE Signal Process Lett. 13 (5), 285–287.
Miri, A., Faez, K., 2018. An image steganography method based on integer wavelet transform. Multimed Tools Appl 77, 13133–13144.
Mitra, S., Dhar, M., Mondal, A., Saha, N., Islam, R., 2015. DCT based Stegano graphic Evaluation parameter analysis in Frequency domain by using modified JPEG luminance Quantization Table. J. Comput. Eng. 17 (1), 68–74.
Morkel, T. Eloff, J.P.H. Olivier M.S., 2005. "An overview of image steganography", In Proceedings of the fifth annual information security South Africa conference (ISSA2005), Sandton, South Africa, June/July (Published electronically).
Patil, P., Bormane, D.S., 2013. DWT based invisible watermarking technique for digital images. Int. J. Eng. Adv. Technol. 2 (4), 603–605.
Popescu, M., Tanasie, R.T., 2012. Graph-Based Volumetric Data Segmentatioın on a Hexagonal-Prismatic Lattice. In: In ProceedIngs of the Federated Conference on Computer Science and Information Sciences, pp. 745–749.
Roy, R., Changder, S., Sarkar, A., Debnath, N.C., 2013. Evaluating Image Steganography Techniques: Future Research Challenges. In: In ProceedIngs of the International Conference on ComputIng, Management and Telecommunications (ComManTel), IEEE, pp. 309–314.
Rustad, S., Setiadi, D.R.I.M., Syukur, A., Andono, P.N., 2022. Inverted LSB image steganography using adaptive pattern to improve imperceptibility. J. King Saud Univers. – Comput. Informat. Sci. 34, 3559–3568.
Sahr, K., 2011. Hexagonal discrete global grid systems for geospatial Computing. Arch. Photogramm., Cartograp. Rem. Sens. 22, 363–376.
Sallee, P., 2003. "Model-Based Steganography", Proceedings of the International Workshop on Digital Watermarking Science. Lecture Notes in Computer Science 2939, 154–167.
Setiadi, D.R.I.M., 2022. Improved payload capacity in LSB image steganography uses dilated hybrid edge detection. J. King Saud Univers. – Comput. Informat. Sci. 34 (2), 104–114. https://doi.org/10.1016/j.jksuci.2019.12.007.
Seyedi, S.H., Aghaeinia, H., Sayadian, A., 2011. A new robust image adaptive steganography method in wavelet domain". In: In ProceedIngs of the 19th Iranian Conference on Electrical EngIneerIng (ICEE), pp. 1–5.
Sharp, T. 2001. "An implementation of key-based digital signal steganography", In Proceedings of 4th international workshop on information hiding, LNCS, vol. 2137, pp. 13–26, Springer.
Shen, S.Y., Huang, L.H., 2015. A data hiding scheme using pixel value differencing and improving exploiting modification directions. Computers & Security 48, 131–141.
Singh, S., Siddiqui, T.J., 2012. Robust image steganography technique based on redundant discrete wavelet transform". In: In ProceedIngs of the 2nd International Conference on Power, Control and Embedded Systems (ICPCES), pp. 1–4.
http://sipi.usc.edu/database/.
Subhedar, M.S., Mankar, V.H., 2014. Current status and key issues in image steganography: A survey. Comput. Sci. Rev. 13–14, 95–113.
Suresh, M., Sam, S.I.S., 2020. Optimized interesting region identification for video steganography using fractional grey wolf optimization along with multiobjective cost function. J. King Saud Univers. – Comput. Informat. Sci. 34, 3489–3496.
H.W. Tseng, H.S. Leng, "A Steganographic Method Based on Pixel-Value Differencing and the Perfect Square Number", J. Appl. Mathemat., vol. 2013, Article ID 189706, 8 pages, 2013. https://doi.org/10.1155/2013/189706.
Wang, Jianjun, Sun, Yiting, Huan, Xu, Chen, Kangkang, Kim, Hyoung Joong, Joo, Sang-Hyun, 2010. An improved section-wise exploiting modification direction method. Signal Process. 90, 2954–2964.
Wu, D.C., Tsai, W.H., 2003. A steganographic method for images by pixel-value differencing. Pattern Recogn. Lett. 24 (9–10), 1613–1626.
Younus, Z.S., Hussain, M.K., 2022. Image steganography using exploiting modification direction for compressed encrypted data. J. King Saud University – Comput. Informat. Sci. 34, 2951–2963.
Younus, Z.S., Younus, G.T., 2020. Video steganography using knight tour algorithm and LSB method for encrypted data. J. Intell. Syst. 29 (1), 1216–1225.
Zhang, S., Su, S., Li, L., Lu, J., Zhou, Q., Chang, C.C., 2022. CSST-Net: an arbitrary image style transfer network of coverless steganography. The Visual Computer 38, 2125–2137.
Zhang, X., Wang, S., Nov. 2006. Efficient steganographic embedding by exploiting modification direction. IEEE Commun. Lett. 10 (11), 781–783.