Multimedia Systems
https://doi.org/10.1007/s00530-020-00703-3

REGULAR PAPER

Image steganography based on Kirsch edge detection

Sudipta Kumar Ghosal[1] · Agneet Chatterjee[2] · Ram Sarkar[2]

Received: 10 August 2020 / Accepted: 29 September 2020

Abstract

Conventional steganography methods fabricate the secret information into the cover pixels without analyzing the pixel intensities of an image. As a result, some minor pixel level manipulations may lead to huge visual distortion in the stego-image. To this end, in this paper, a novel steganographic scheme based on Kirsch edge detector is proposed. The aim of the scheme is to maximize the payload by embedding more secret bits into edge pixels and fewer bits into the non-edge pixels. The proposed scheme has three major phases: construction of edge image, embedding and extraction. The first phase deals with the construction of masked image from the cover image, and in turn, edge image from the masked one. The second phase deals with the decomposition of the cover image into a set of triplet of pixels and then embedding of $(x+y+1)$ bits of secret data into each triplet of pixels to obtain the stego-image. Here, 'x' and 'y' are not fixed as the edge information of each triplet changes incessantly. The third or last phase deals with the extraction of the secret information from the stego-image using the reverse process. Simulation results on some standard images ensure that the proposed method achieves higher payload and better image quality compared to the conventional steganographic schemes. Furthermore, the Kirsch edge detector is able to produce more number of edge pixels compared to the traditional edge detectors; and, hence the proposed scheme also outperforms the existing edge-based methods in terms of payload.

Keywords Steganography · Edge detector · Kirsch · Payload · Image quality

1 Introduction

With the rapid growth of image contents, the challenge of maintaining privacy of such contents over the internet has become a major concern in the recent times. The intruders are always try to access the useful image content using efficient image retrieval algorithms [1] and then launch a surprise attack in the form of image malware. Using these images and deep learning models, machines can accurately identify and classify objects and then explore several applications in the field of computer vision [2, 3]. Two security approaches are adopted by the researcher to defend the intruders from accessing those contents: Cryptography and Steganography. Cryptography, the process of transforming confidential data into non-readable form, provides secure data transfer although the encrypted form of data attracts attention and reveals its importance when intercepted. On the other hand, the research aiming at hiding information in an image to accomplish the imperceptibility is referred to as steganography. The prime goal of any steganography method, unlike in cryptography, is to hide the very presence of concealed information and focus most on high payload, good imperceptibility and substantial robustness. The purpose of such a goal is to exploit the weakness of the Human Visual System (HVS) against detecting even a minimal change in an image's distortion. Usually, an image steganographic scheme works either in spatial domain or in transform domain. Spatial domain methods ensure the fabrication of secret information through pixel level manipulation while the transform domain methods convert the cover images into

Communicated by Y. Zhang.

Sudipta Kumar Ghosal
sudipta.ghosal@gmail.com

Agneet Chatterjee
agneet257@gmail.com

Ram Sarkar
ramjucse@gmail.com

[1] Department of Computer Science and Technology, Nalhati Government Polytechnic, Birbhum, Nalhati 731243, India

[2] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

Published online: 12 October 2020

transform domain and then insert the secret information through coefficient level manipulations.

Most of the spatial domain steganographic techniques, which work on the pixel level, suffer from embedding constant payload of data in each pixel block. One such technique is the Least Significant Bit (LSB) substitution [4] which aims at replacing the least significant bit of each pixel with the corresponding hidden data pixel. As a result, such a pixel changes its original magnitude by 0 or 1, a trend which is cascaded throughout the cover image. In the year 2003, Wu et al. [5] introduced the concept of Pixel Value Differencing (PVD) where the differences of successive pair of pixels are calculated and then checked range in which they fall against a predefined set of values. The larger difference signifies that the pixel is “edge” and the smaller difference signifies that the pixel is “smooth”. Due to the HVS model, the alteration made into the edge pixels reflects higher order distortion than the alteration made into the smooth pixels. This implies that it is advantageous to hide more data into the pixels which are considered as the edge pixels. Such a method allows for non-uniform embedding of data; thereby, increasing robustness and payload, and takes into consideration the imperceptibility. Couple of years later, Zhang and Wang [6] suggested an effective steganographic scheme named Exploiting Modification Direction (EMD),where the hidden information is considered to be a $(2n+1)$-ary secret digit and the same is concealed into n pixels of the carrier image. The major benefit of EMD method over LSB method is that n bits of secret data can be embedded into n cover pixels by altering only one pixel from a set of $(2n+1)$ combination. In 2007, Lee et al. [7] proposed an improved embedding scheme based on the existing EMD scheme in order to increase the payload. This scheme uses an 8-ary notational system and achieved a payload of 1.5 times higher than the former one. Few years later, Shen et al. [8] combined these two methods and proposed a novel steganographic method which determines the embedding of base digits from each pair of pixels for EMD-based embedding process. In 2017, they also proposed an interpolation-based steganographic scheme [9] which can offer high payload, reduced computational complexity and acceptable visual clarity. In 2019, Mukherjee and Sanyal [10] proposed an adaptive steganographic method that used Power Modulus Scrambling (PMS). The scheme achieved high security using block-based pixel swapping. The secret information is fabricated on the basis of the key-based permutation combination. Subsequently, a two-step embedding process is proposed by Abdullah et al. [11] in the same year. In first step, it dealt with manipulation of cover and secret images before embedding and in the second step, it achieved higher 0:1 ratio in both hidden bit stream and LSB plane of the cover image.

The aforementioned steps have been exploited to use a bit-plane mapping instead of bit-plane replacement to embed the secret data. In 2020, Chatterjee et al. [12] proposed an optical character recognition (OCR)-based LSB method in which the image level features from the characters of the textual message are extracted, and then fabricated the same within the cover image. The scheme is extremely effective for secret images which contain textual information only. Later this year, Mohamed et al. [13] proposed a quad-tree-based steganographic scheme in $L^*a^*b^*$ color space where the cover image is segmented into high and low correlated adaptive-size blocks. The embedding is done on the high frequency regions of the discrete cosine transform (DCT) which belongs to the highly correlated blocks to ensure high payload and visual clarity.

Apart from methods mentioned above, over the last few years, traditional edge detection methods such as Sobel, Prewitt and Canny etc. have been used in the field of image steganography. Chen et al. [14] and Tseng et al. [15] are the pioneers of edge-based steganography methods. Both the said methods are focussed at spreading out the hidden data embedding throughout the cover image depending on the nature of the pixels. However, the more concern is given to the metric payload as the conventional edge detectors produce limited numbers of edge pixels from an image. Further, there is a burden of embedding extra edge information for data extraction purpose. In 2017, Bai et al. [16] designed a steganography technique that accommodated a larger payload by utilizing the popular edge detectors such as Sobel, Canny and Fuzzy. This scheme not only outperformed the former methods [13, 14] in terms of payload but also it avoided the problem of embedding extra edge information. Within a year, Atta et al. [17] introduced an edge detector based neutroscopic set (NSED) in the steganography domain that exploited wavelet packet decomposition (WPD) for embedding more secret bits into the coefficients of the edge tree than the non-edge tree. In the same year, Gaurav and Ghanekar [18] presented a novel steganography scheme based on the local reference edge detection technique and XOR disjunction property. In this method, the secret information is fabricated in the sharp regions by local reference pixels as obtained by Canny edge detector and morphological operator. The embedding process ensured both high security and high capacity based on bit plane-dependent XOR coding process that makes minimal possible alteration in the LSB bits of the edge pixels. Taking inspiration from these methods, Ghosal et al. [19] also investigated Laplacian of Gaussian (LoG) operator in the steganography domain, and found better outcomes than many of the former methods. In 2019, Setiadi [20] introduced a dilated hybrid edge detection-based steganographic scheme, where they combined the output edge images of Canny and Sobel edge detectors using bitwise-OR operation. The simulation results ensured higher payload with reduced quality degradation in the stego-image. Later in 2019, Kumar et al. [21] proposed an adaptive steganography method based on the novel fuzzy edge detection for estimating the precise edge areas of a cover image. Secret bits are fabricated into the edge pixels to obtain minimal distortion in the stego-images. Considering the pros and cons of all the aforementioned methods, we can conclude that both traditional and edge-based methods achieve good imperceptibility although most of them suffers due to low payload. Hence, in order to identify more accurate places of edges in the images which help in maximizing the payload, a novel steganography based on Kirsch edge detection [22] is proposed in this paper. The performance of the proposed scheme is superior to the state-of-the-art methods that is evident from the data shown in the result section.

The organization of the paper is as follows: Sect. 2 highlights the motivation and contributions of this work, whereas Sect. 3 gives a brief overview of Kirsch edge detection technique. Proposed scheme is explained in Sect. 4. Experimental results, analysis and the discussion are reported in Sect. 5 and the security analysis part is elaborated in Sect. 6. Lastly, conclusion along with some future plans is specified in Sect. 7.

2 Motivation and contributions

In Sect. 1, we have discussed the problems associated with conventional and edge-based steganography methods. Mostly, these methods attain less payload, high visual distortion and low robustness. From the literature survey, it is clear that the distortion occurring in image content due to specific attack is much lesser in edge areas as compared to the non-edge areas. Hence, many edge detection techniques, such as Sobel, Prewitt, Canny and Laplacian of Gaussian etc. have been adopted in the field of image steganography in recent times. However, these edge detectors sometimes fail to achieve satisfactorily results mainly because they produce less number of edges or generate some noises. In contrast, Kirsch edge detector generates thicker edges, i.e. it produces more edge pixels compared to the conventional edge detectors. The reason is the Kirsch operator involves a larger numbers of pixels in either direction and assigns higher weights. As a result, the cover pixels are exploited efficiently and achieved payload of the proposed method is found to be better than the earlier methods.

In a nutshell, the proposed Kirsch edge detector-based steganographic method has the following key advantages:

- o The proposed method supports widely used image formats such as BMP, PPM, PGM and TIFF.
- o Kirsch edge detector produces more edge pixels than conventionally used methods, such as Canny, Sobel and Prewitt.
- o More edge pixels ensure a variable payload in the proposed scheme (up to 3 bpp).
- o The quality of the stego-image never becomes less than the acceptable level which can be verified from results, measured in terms of some standard metrics, provided in Sect. 5.
- o The stego-images have been tested using the StegExpose tool in Sect. 6 and it is found that the robustness is very high.

3 Kirsch edge detector: an overview

Kirsch operator (also known as Kirsch compass kernel) [22] is a linear edge detector that detects the maximum edge strength in eight predetermined directions. Here, the single mask is used to rotate with 45 degrees in the eight directions—North (N), North West (NW), West (W), South West (SW), South (S), South East (SE), East (E) and North East (NE). The basic advantage of this approach is that a good number of effective edges are found in each of these masks. For the sake of clarity, let us consider the 3×3 sub-matrix of an image as:

$$\begin{bmatrix} p_3 & p_2 & p_1 \\ p_4 & p_c & p_8 \\ p_5 & p_6 & p_7 \end{bmatrix}$$

With reference to the pixel located in the centre, i.e. p_c, the contrast function is evaluated as:

$$\max\left[1, \max_{i=0}^{7}\left|5\left(p_i + p_{i+1} + p_{i+2}\right) - 3\left(p_{i+3} + p_{i+4} + \cdots + p_{i+7}\right)\right|\right]$$

where the subscripts are ranged from 0 to 7. The edge masks are given below:

$$\mathrm{N} = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad \mathrm{NW} = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$\mathrm{W} = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 3 & -3 \end{bmatrix} \quad \mathrm{SW} = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$$

$$\mathrm{S} = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad \mathrm{SE} = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

$$\mathrm{E} = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad \mathrm{NE} = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

It is observed that the aforementioned non-isotropic function is related to the magnitude of the gradient of the said function. It is non-symmetric and very sensitive to minor changes in the value of the gradient.

4 Proposed technique

The novelty of the proposed steganographic scheme lies in the usage of Kirsch edge detection technique to achieve high payload at the cost of minimal quality distortion of the stego-image as Kirsch edge detector yields more edge pixels compared to conventionally used edge detectors mentioned earlier. In our proposed method, Kirsch operator primarily classifies the cover image into edge and non-edge pixels. We know that tolerance level of an edge pixel against noise is higher than a non-edge pixel as far as the HVS is concerned. Therefore, the cover image is decomposed and the secret bits are embedded into the pixels in such a way that more secret bits get embedded into the edge pixels in comparison with the non-edge pixels.

The proposed technique is elaborated in the following subsections. Here, the construction of edge image, embedding and extraction processes are described in Sects. 4.1, 4.2 and 4.3, respectively.

4.1 Construction of edge image

To find and preserve identical edges during embedding and extraction phases, Kirsch operator is applied over a masked image (I_{MASK}) instead of the cover image (I_{C}). In general, we know that the 8-bit gray-scale image consists of 8-bit planes: the 1st bit plane contains the set of the LSBs (LSB-0), the 2nd bit plane contains the set of the pre-LSBs (LSB-1) and so on. This concept inspired us to construct a masked image (I_{MASK}) by setting bit planes 1–7 to zero. These alterations do not affect the MSB bit plane, i.e. bit plane—8 of I_{C}. Further, despite losing bit information, it is examined that I_{MASK} is able to construct an edge image ($I_{\text{MASK-EDGE}}$) very similar to the edge image (I_{EDGE}) obtained without losing 7-LSBs. The embedding phase, to be discussed in the next subsection, ensures that in order to obtain maximum payload, alteration should not be made at the MSB of each pixel.

As discussed in Sect. 3, Kirsch operator produces separate edge images corresponding to eight different directions and hence, maximum edge strength in all eight directions are considered to construct the final output edge image. Figure 1 shows the cover image (I_{C}), masked image (I_{MASK}), edge mages obtained in all eight directions—N, NW, W, SW, S, SE, E and NE ($I_{\text{MASK-N-EDGE}}$, $I_{\text{MASK-NW-EDGE}}$, $I_{\text{MASK-W-EDGE}}$, $I_{\text{MASK-SW-EDGE}}$, $I_{\text{MASK-S-EDGE}}$, $I_{\text{MASK-SE-EDGE}}$, $I_{\text{MASK-E-EDGE}}$, $I_{\text{MASK-NE-EDGE}}$) and the final masked edge image ($I_{\text{MASK-EDGE}}$), respectively.

4.2 Embedding

This section deals with the embedding process in detail. Here, the cover image I_{C} is divided into a set of triplet of pixels (p_i, p_{i+1}, p_{i+2}) in row major order. To determine the embedding parameters x and y for both Case 1 and Case 2, eight possible edge combinations obtained from $I_{\text{MSB-EDGE}}$ have been considered as shown in Table 1.

In this process, the pixels of each triplet (p_i, p_{i+1}, p_{i+2}) have been shuffled to identify effective embedding ratio and are saved as q_i, q_{i+1} and q_{i+2}, respectively. A reference value f is obtained as follows:

$$f = \left(q_i + 2^x q_{i+1} + 2^y q_{i+2}\right)\%2^{(x+y+1)}$$

A decimal number s is constructed by retrieving $(x+y+1)$-bits from the secret bit-stream and then a difference value d is obtained as $d=(s-f)$. The variable d usually belongs to the range [0, $2^{x+y+1}-1$] but, to reduce deviation of pixel values followed by embedding, we always keep d in the range [-2^{x+y}, 2^{x+y}] by applying a local adjustment as follows:

$$d = \begin{cases} d + 2^{(x+y+1)} : \text{if } d < -2^{x+y} \\ d - 2^{(x+y+1)} : \text{if } d > 2^{x+y} \end{cases}$$

Further, the sign ∈ of the modified difference value d is determined as follows:

$$\in = \begin{cases} -1 : \text{if } d < 0 \\ 1 : \text{otherwise} \end{cases}$$

The non-negative d value (i.e. abs(d)) is decomposed into three decimal segments α, β and γ as follows:

$$\alpha = \frac{\text{abs}(d)}{2^y}$$

$$\text{temp} = \text{abs}(d)\%2^y$$

$$\beta = \frac{\text{temp}}{2^x}$$

$$\gamma = \text{temp}\%2^y$$

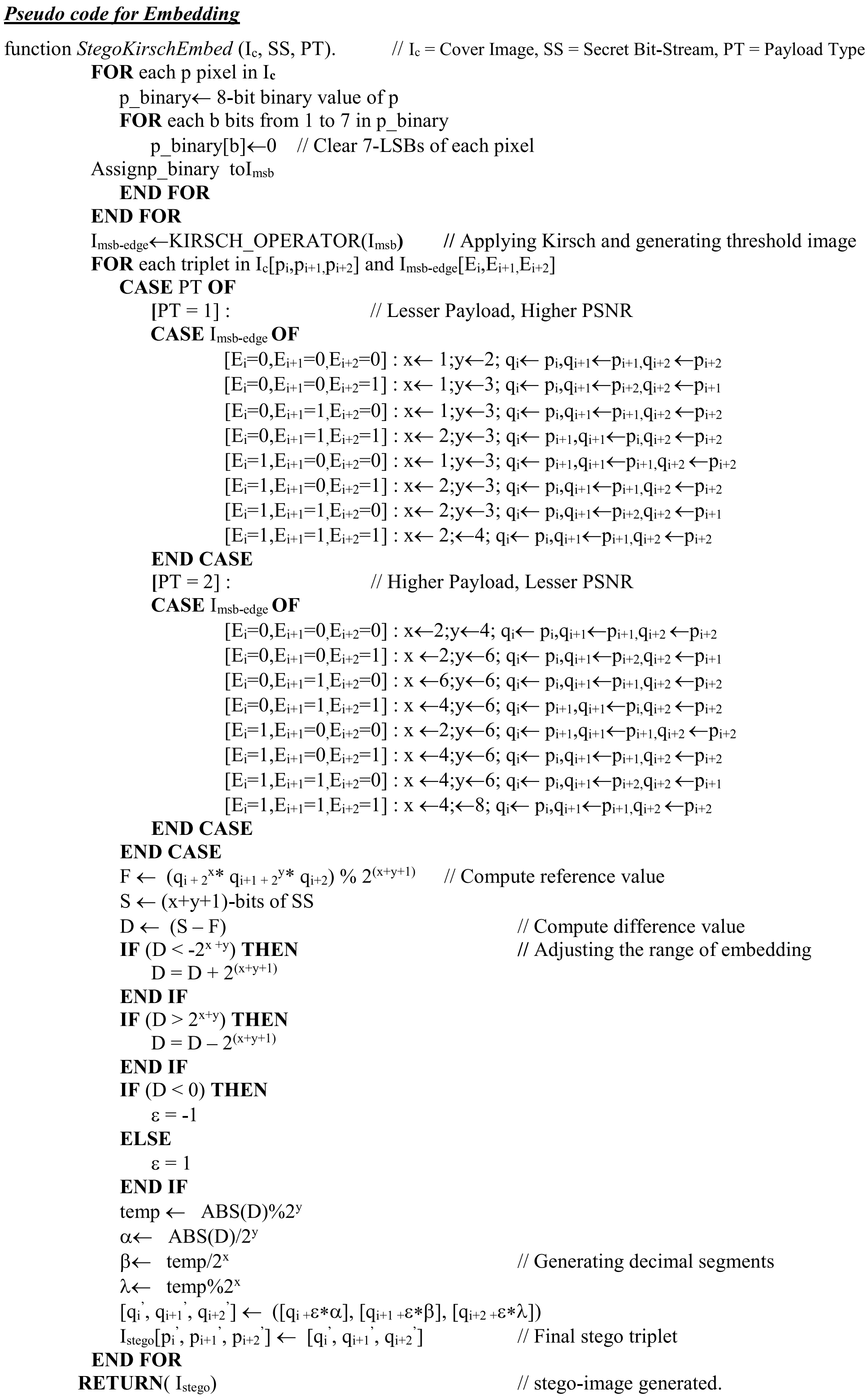
Here, temp denotes the temporary variable for intermediate calculation.

Three decimal segments α, β and γ are then added to the shuffled pixels q_i, q_{i+1} and q_{i+2} to obtain the shuffled stego-pixels $q'_{i'}$, q'_{i+1} and q'_{i+2}as follows:

$$[q'_i, q'_{i+1}, q'_{i+2}] = [(q_i + \epsilon \times \gamma), \left(q_{i+1} + \epsilon \times \beta\right), (q_{i+2} + \epsilon \times \alpha)]$$

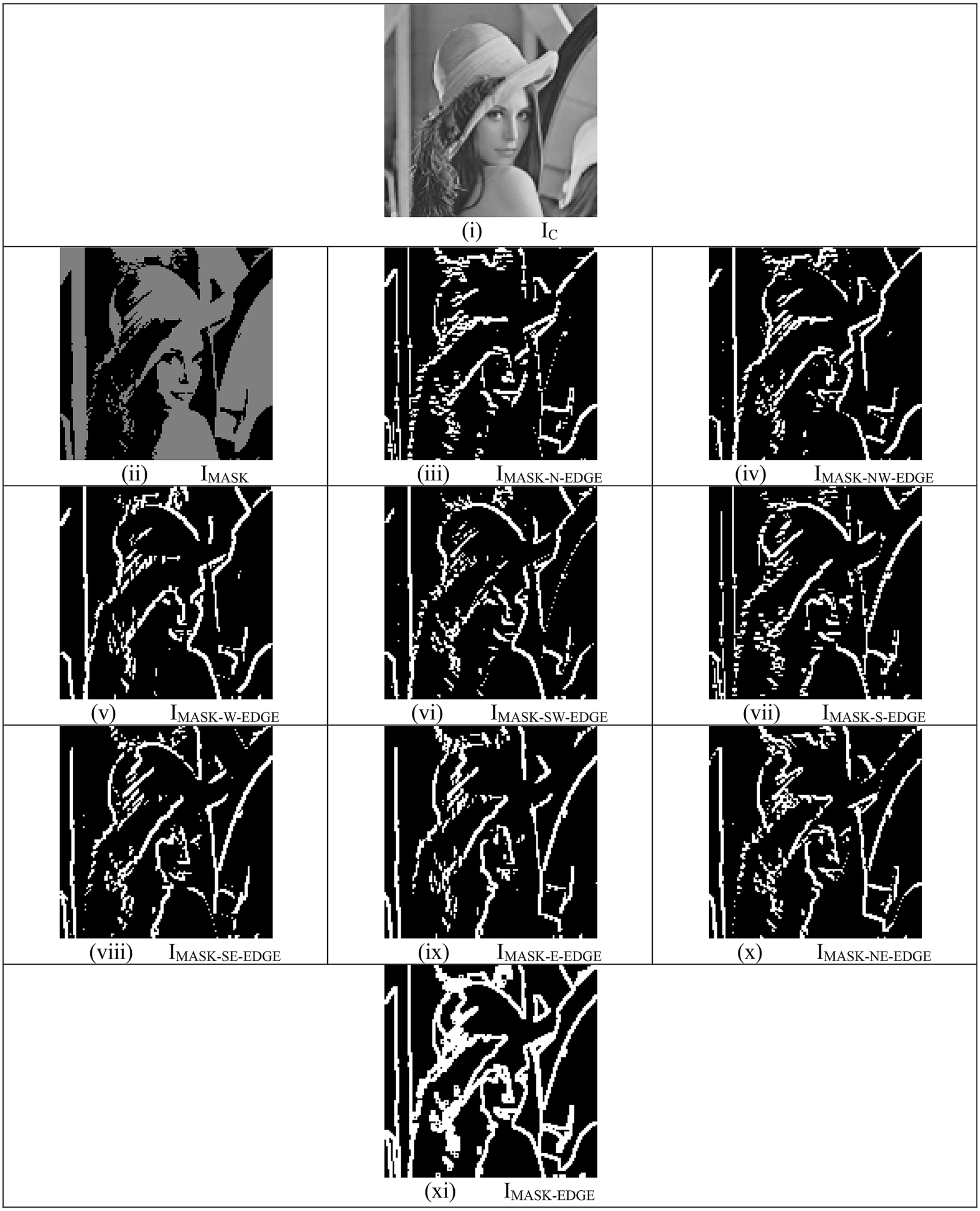
Finally, the order of the stego-pixels ($q'_{i'}$, q'_{i+1}, q'_{i+2}) is restored back based on the edge information to obtain output stego-pixels ($p'_{i'}$, p'_{i+1}, p'_{i+2}). This process is repeated until the entire secret bit-stream is concealed and the final stego-image I_O is produced.

Figure 2 depicts the flow diagram of the embedding process in great detail.



Pseudo code for Embedding

```
function StegoKirschEmbed ($I_c$, SS, PT).          // $I_c$ = Cover Image, SS = Secret Bit-Stream, PT = Payload Type
            FOR each p pixel in $I_c$
                p_binary← 8-bit binary value of p
                FOR each b bits from 1 to 7 in p_binary
                    p_binary[b]←0   // Clear 7-LSBs of each pixel
            Assignp_binary  to$I_{msb}$
                END FOR
            END FOR
            $I_{msb\text{-}edge}$←KIRSCH_OPERATOR($I_{msb}$)       // Applying Kirsch and generating threshold image
            FOR each triplet in $I_c[p_i,p_{i+1},p_{i+2}]$ and $I_{msb\text{-}edge}[E_i,E_{i+1},E_{i+2}]$
                CASE PT OF
                    [PT = 1] :                    // Lesser Payload, Higher PSNR
                    CASE $I_{msb\text{-}edge}$ OF
                              [$E_i$=0,$E_{i+1}$=0,$E_{i+2}$=0] : x← 1;y←2; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1}$,$q_{i+2}$ ←$p_{i+2}$
                              [$E_i$=0,$E_{i+1}$=0,$E_{i+2}$=1] : x← 1;y←3; $q_i$← $p_i$,$q_{i+1}$←$p_{i+2}$,$q_{i+2}$ ←$p_{i+1}$
                              [$E_i$=0,$E_{i+1}$=1,$E_{i+2}$=0] : x← 1;y←3; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1}$,$q_{i+2}$ ←$p_{i+2}$
                              [$E_i$=0,$E_{i+1}$=1,$E_{i+2}$=1] : x← 2;y←3; $q_i$← $p_{i+1}$,$q_{i+1}$←$p_i$,$q_{i+2}$ ←$p_{i+2}$
                              [$E_i$=1,$E_{i+1}$=0,$E_{i+2}$=0] : x← 1;y←3; $q_i$← $p_{i+1}$,$q_{i+1}$←$p_{i+1}$,$q_{i+2}$ ←$p_{i+2}$
                              [$E_i$=1,$E_{i+1}$=0,$E_{i+2}$=1] : x← 2;y←3; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1}$,$q_{i+2}$ ←$p_{i+2}$
                              [$E_i$=1,$E_{i+1}$=1,$E_{i+2}$=0] : x← 2;y←3; $q_i$← $p_i$,$q_{i+1}$←$p_{i+2}$,$q_{i+2}$ ←$p_{i+1}$
                              [$E_i$=1,$E_{i+1}$=1,$E_{i+2}$=1] : x← 2;←4; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1}$,$q_{i+2}$ ←$p_{i+2}$
                    END CASE
                    [PT = 2] :                    // Higher Payload, Lesser PSNR
                    CASE $I_{msb\text{-}edge}$ OF
                              [$E_i$=0,$E_{i+1}$=0,$E_{i+2}$=0] : x←2;y←4; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1}$,$q_{i+2}$ ←$p_{i+2}$
                              [$E_i$=0,$E_{i+1}$=0,$E_{i+2}$=1] : x ←2;y←6; $q_i$← $p_i$,$q_{i+1}$←$p_{i+2}$,$q_{i+2}$ ←$p_{i+1}$
                              [$E_i$=0,$E_{i+1}$=1,$E_{i+2}$=0] : x ←6;y←6; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1}$,$q_{i+2}$ ←$p_{i+2}$
                              [$E_i$=0,$E_{i+1}$=1,$E_{i+2}$=1] : x ←4;y←6; $q_i$← $p_{i+1}$,$q_{i+1}$←$p_i$,$q_{i+2}$ ←$p_{i+2}$
                              [$E_i$=1,$E_{i+1}$=0,$E_{i+2}$=0] : x ←2;y←6; $q_i$← $p_{i+1}$,$q_{i+1}$←$p_{i+1}$,$q_{i+2}$ ←$p_{i+2}$
                              [$E_i$=1,$E_{i+1}$=0,$E_{i+2}$=1] : x ←4;y←6; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1}$,$q_{i+2}$ ←$p_{i+2}$
                              [$E_i$=1,$E_{i+1}$=1,$E_{i+2}$=0] : x ←4;y←6; $q_i$← $p_i$,$q_{i+1}$←$p_{i+2}$,$q_{i+2}$ ←$p_{i+1}$
                              [$E_i$=1,$E_{i+1}$=1,$E_{i+2}$=1] : x ←4;←8; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1}$,$q_{i+2}$ ←$p_{i+2}$
                    END CASE
                END CASE
                $F \leftarrow (q_{i+2}{}^{x} * q_{i+1+2}{}^{y} * q_{i+2}) \% 2^{(x+y+1)}$      // Compute reference value
                $S \leftarrow (x+y+1)$-bits of SS
                $D \leftarrow (S - F)$                                  // Compute difference value
                IF ($D < -2^{x+y}$) THEN                               // Adjusting the range of embedding
                    $D = D + 2^{(x+y+1)}$
                END IF
                IF ($D > 2^{x+y}$) THEN
                    $D = D - 2^{(x+y+1)}$
                END IF
                IF (D < 0) THEN
                    $\varepsilon = -1$
                ELSE
                    $\varepsilon = 1$
                END IF
                temp ←  ABS(D)%$2^y$
                $\alpha$←  ABS(D)/$2^y$
                $\beta$←  temp/$2^x$                                     // Generating decimal segments
                $\lambda$←  temp%$2^x$
                $[q_i', q_{i+1}', q_{i+2}'] \leftarrow ([q_i + \varepsilon*\alpha], [q_{i+1} + \varepsilon*\beta], [q_{i+2} + \varepsilon*\lambda])$
                $I_{stego}[p_i', p_{i+1}', p_{i+2}'] \leftarrow [q_i', q_{i+1}', q_{i+2}']$           // Final stego triplet
            END FOR
        RETURN( $I_{stego}$)                                    // stego-image generated.
```



(i) I_C

(ii) I_{MASK}

(iii) $I_{MASK\text{-}N\text{-}EDGE}$

(iv) $I_{MASK\text{-}NW\text{-}EDGE}$

(v) $I_{MASK\text{-}W\text{-}EDGE}$

(vi) $I_{MASK\text{-}SW\text{-}EDGE}$

(vii) $I_{MASK\text{-}S\text{-}EDGE}$

(viii) $I_{MASK\text{-}SE\text{-}EDGE}$

(ix) $I_{MASK\text{-}E\text{-}EDGE}$

(x) $I_{MASK\text{-}NE\text{-}EDGE}$

(xi) $I_{MASK\text{-}EDGE}$

Fig. 1 Cover, masked and different edges of masked ‘Lena’ image obtained by through Kirsch edge detector

4.3 Extraction

As discussed in Sect. 4.1, $I_{\text{MSB-EDGE}}$ can be re-computed from I_{O}. The present section elaborates the extraction process of secret information from I_{O}. Similar to the embedding process, the stego-image I_{O} is divided into a set of triplet of pixels (p_i, p_{i+1}, p_{i+2}) in row major order. To determine the extraction parameters x and y, all eight edge combinations, as shown in Table 1, have been considered.

Pseudo code for Extraction

```
function StegoKirschExtract ($I_o$, PT).          // $I_o$ = Stego-image, PT = Payload Type
            FOR each p pixel in $I_o$
                p_binary← 8-bit binary value of p
                FOR each b bits from 1 to 7 in p_binary
                    p_binary[b]←0    // Clear 7-LSBs of each pixel
            Assignp_binary  to$I_{msb}$
                END FOR
            END FOR
            S = []                                          // Output Secret Stream
            $I_{msb-edge}$←KIRSCH_OPERATOR($I_{msb}$)        // Applying Kirsch and generating threshold image
            FOR each triplet in $I_o$[$p_i$,$p_{i+1,}$$p_{i+2}$] and $I_{msb-edge}$[$E_i$,$E_{i+1,}$$E_{i+2}$]
                CASE PT OF
                    [PT = 1] :                     //  C1

               CASE $I_{msb-edge}$ OF
                         [$E_i$=0,$E_{i+1}$=0,$E_{i+2}$=0] : x← 1;y←2; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1,}$$q_{i+2}$ ←$p_{i+2}$
                         [$E_i$=0,$E_{i+1}$=0,$E_{i+2}$=1] : x← 1;y←3; $q_i$← $p_i$,$q_{i+1}$←$p_{i+2,}$$q_{i+2}$ ←$p_{i+1}$
                         [$E_i$=0,$E_{i+1}$=1,$E_{i+2}$=0] : x← 1;y←3; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1,}$$q_{i+2}$ ←$p_{i+2}$
                         [$E_i$=0,$E_{i+1}$=1,$E_{i+2}$=1] : x← 2;y←3; $q_i$← $p_{i+1}$,$q_{i+1}$←$p_{i,}$$q_{i+2}$ ←$p_{i+2}$
                         [$E_i$=1,$E_{i+1}$=0,$E_{i+2}$=0] : x← 1;y←3; $q_i$← $p_{i+1}$,$q_{i+1}$←$p_{i+1,}$$q_{i+2}$ ←$p_{i+2}$
                         [$E_i$=1,$E_{i+1}$=0,$E_{i+2}$=1] : x← 2;y←3; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1,}$$q_{i+2}$ ←$p_{i+2}$
                         [$E_i$=1,$E_{i+1}$=1,$E_{i+2}$=0] : x← 2;y←3; $q_i$← $p_i$,$q_{i+1}$←$p_{i+2,}$$q_{i+2}$ ←$p_{i+1}$
                         [$E_i$=1,$E_{i+1}$=1,$E_{i+2}$=1] : x← 2;←4; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1,}$$q_{i+2}$ ←$p_{i+2}$
               END CASE
               [PT = 2] :                     // C2
               CASE $I_{msb-edge}$ OF
                         [$E_i$=0,$E_{i+1}$=0,$E_{i+2}$=0] : x←2;y←4; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1,}$$q_{i+2}$ ←$p_{i+2}$
                         [$E_i$=0,$E_{i+1}$=0,$E_{i+2}$=1] : x ←2;y←6; $q_i$← $p_i$,$q_{i+1}$←$p_{i+2,}$$q_{i+2}$ ←$p_{i+1}$
                         [$E_i$=0,$E_{i+1}$=1,$E_{i+2}$=0] : x ←6;y←6; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1,}$$q_{i+2}$ ←$p_{i+2}$
                         [$E_i$=0,$E_{i+1}$=1,$E_{i+2}$=1] : x ←4;y←6; $q_i$← $p_{i+1}$,$q_{i+1}$←$p_{i,}$$q_{i+2}$ ←$p_{i+2}$
                         [$E_i$=1,$E_{i+1}$=0,$E_{i+2}$=0] : x ←2;y←6; $q_i$← $p_{i+1}$,$q_{i+1}$←$p_{i+1,}$$q_{i+2}$ ←$p_{i+2}$
                         [$E_i$=1,$E_{i+1}$=0,$E_{i+2}$=1] : x ←4;y←6; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1,}$$q_{i+2}$ ←$p_{i+2}$
                         [$E_i$=1,$E_{i+1}$=1,$E_{i+2}$=0] : x ←4;y←6; $q_i$← $p_i$,$q_{i+1}$←$p_{i+2,}$$q_{i+2}$ ←$p_{i+1}$
                         [$E_i$=1,$E_{i+1}$=1,$E_{i+2}$=1] : x ←4;←8; $q_i$← $p_i$,$q_{i+1}$←$p_{i+1,}$$q_{i+2}$ ←$p_{i+2}$
               END CASE
           END CASE
           F ←  ($q_{i + 2}{}^{x}$* $q_{i+1 + 2}{}^{y}$* $q_{i+2}$) % $2^{(x+y+1)}$     // Compute reference value
           S ←S + BINARY(F) [x+y+1]-bits
        END FOR
      RETURN (S)                                    // Output Secret Stream
```

Similar to the embedding phase, the pixels of each triplet (p_i, p_{i+1}, p_{i+2}) have been shuffled and are saved as q_i, q_{i+1} and q_{i+2}, respectively. A reference value f is obtained as follows:

$$f = \left(q_i + 2^x q_{i+1} + 2^y q_{i+2}\right)\%2^{(x+y+1)}$$

The obtained reference value f is nothing but the decimal equivalent corresponding to the $(x+y+1)$-bits of extracted secret information. Successive steps ensure the extraction of entire secret bit-stream.

Figure 3 depicts the flow diagram of the extraction process in great detail.

5 Experimental results, analysis and discussion

For evaluation of the proposed steganography method, the experiments are performed on 10 benchmark gray scale images [23] of dimension 129 × 129 as shown in Fig. 4. Three standard evaluation metrics are used to assess the

Table 1 Determination of bit embedding parameters based on the edge combination of each triplet of pixels for a cover image

Edge combination w.r.t. original triplet of pixels (p_i, p_{i+1}, p_{i+2})			Shuffled triplet of pixels			Bit embedding parameter			
						Case 1 (C1)		Case 2 (C2)	
E_i	E_{i+1}	E_{i+2}	q_i	q_{i+1}	q_{i+2}	x	y	x	y
0	0	0	p_i	p_{i+1}	p_{i+2}	1	2	2	4
0	0	1	p_i	p_{i+2}	p_{i+1}	1	3	2	6
0	1	0	p_i	p_{i+1}	p_{i+2}	1	3	2	6
0	1	1	p_{i+1}	p_i	p_{i+2}	2	3	4	6
1	0	0	p_{i+1}	p_i	p_{i+2}	1	3	2	6
1	0	1	p_i	p_{i+1}	p_{i+2}	2	3	4	6
1	1	0	p_i	p_{i+2}	p_{i+1}	2	3	4	6
1	1	1	p_i	p_{i+1}	p_{i+2}	2	4	4	8

strength of the proposed scheme: payload, stego-image's quality and robustness. Payload is measured in terms of bits per pixel (bpp) while the Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) are the metrics to analyze the quality of the stego-images. The PSNR (in dB) is defined as follows:

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{\mathrm{MAX}_I}{\sqrt{\mathrm{MSE}}}\right) \tag{1}$$

Here, $\mathrm{MAX_I}$ is 255 as each pixel of a gray-scale image (I) is represented by 8-bits only. However, the metric MSE is expressed as the average squared difference between each original pixel and its distorted version. It is computed as the summation of the squared differences of all the pixels followed by dividing it with the total number of pixels. The MSE can be defined as follows:

$$\mathrm{MSE} = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[I_c(i,j) - I_o(i,j)\right]^2 \tag{2}$$

Here, both cover image (I_C) and the stego-image (I_O) must have identical dimension, i.e. $M \times N$ as assumed here.

The SSIM is used to measure the similarity in the structure of the two images. The visual quality of the watermarked image with respect to the original image can be measured in terms of SSIM. The structural similarity value lies between − 1 and 1. A value close to 1 indicates the higher similarity. It is mathematically expressed as follows:

$$\mathrm{SSIM}\left(I_C, I_O\right) = \frac{\left(2\mu_{I_C}\mu_{I_O} + c_1\right)\left(2\sigma_{I_C.I_O} + c_2\right)}{\left(\mu_{I_C}^2 + \mu_{i_O}^2 + c_1\right)\left(\sigma_{CI}^2 + \sigma_{I_O}^2 + c_2\right)}$$

where, μ_{I_C}, μ_{I_O}, σ_{I_C}, σ_{I_O} and $\sigma_{I_C.I_O}$ are mean of I_C and I_O, variance of I_C and I_O, the covariance of I_C and I_O, respectively. Here, c_1 and c_2 are constants which are used to stabilize the division with weak denominator.

In general, a better steganography algorithm should have high payload along with good quality of stego-image, but one has to be careful that under no circumstance, the value of PSNR gets less than 30 dB.

Since, the cover image is decomposed into 1 × 3 block, the benchmark images of dimension 128 × 128 has been resized to 129 × 129 by padding a duplicate row and column. In Table 2, two different sets of results (i.e. cases C1 and C2) have been summarized. The results of six important metrics such as no. of edge pixels, payload (bpp), PSNR (dB), SSIM, MSE and UQI have been taken into consideration. The first metric is identical for both cases of an image, however the remaining metrics vary. Here, 'Baboon' image ensured highest number of edge pixels (i.e. 8588) and hence, the payload values are maximal among the obtained results [i.e. 1.8494 bpp and 3.3654 bpp]. For Case 1, PSNR, SSIM, MSE and UQI values are in the range [44.90–48.47 dB], [0.9952–0.9982], [0.92–2.09] and [0.9815–0.9999], respectively. Similarly, for Case 2, PSNR, SSIM, MSE and UQI values are in the range [31.81–37.35 dB], [0.9252–0.9794], [11.95–42.78] and [0.9105–0.9716], respectively. However, the average case results ensure that the payload is ~ 3bpp, PSNR is more than 34 dB, SSIM is 0.9619 and UQI is 0.9258. The PSNR value above 30 dB is considered as acceptable image quality and the SSIM and UQI values close to one indicates minimal structural difference and pixel independence involving cover and stego-images.

Tables 3 and 4 shows that the number of edge pixels obtained through the Kirsch edge detector-based proposed steganography scheme is relatively high as compared to that of Canny or LoG edge detectors. It results in significant payload improvement at the cost of considerable quality distortion. Considering the results obtained on 10 benchmark images as shown in Fig. 4, the number of edge pixels for Canny or LoG edge detectors are relatively low as compared

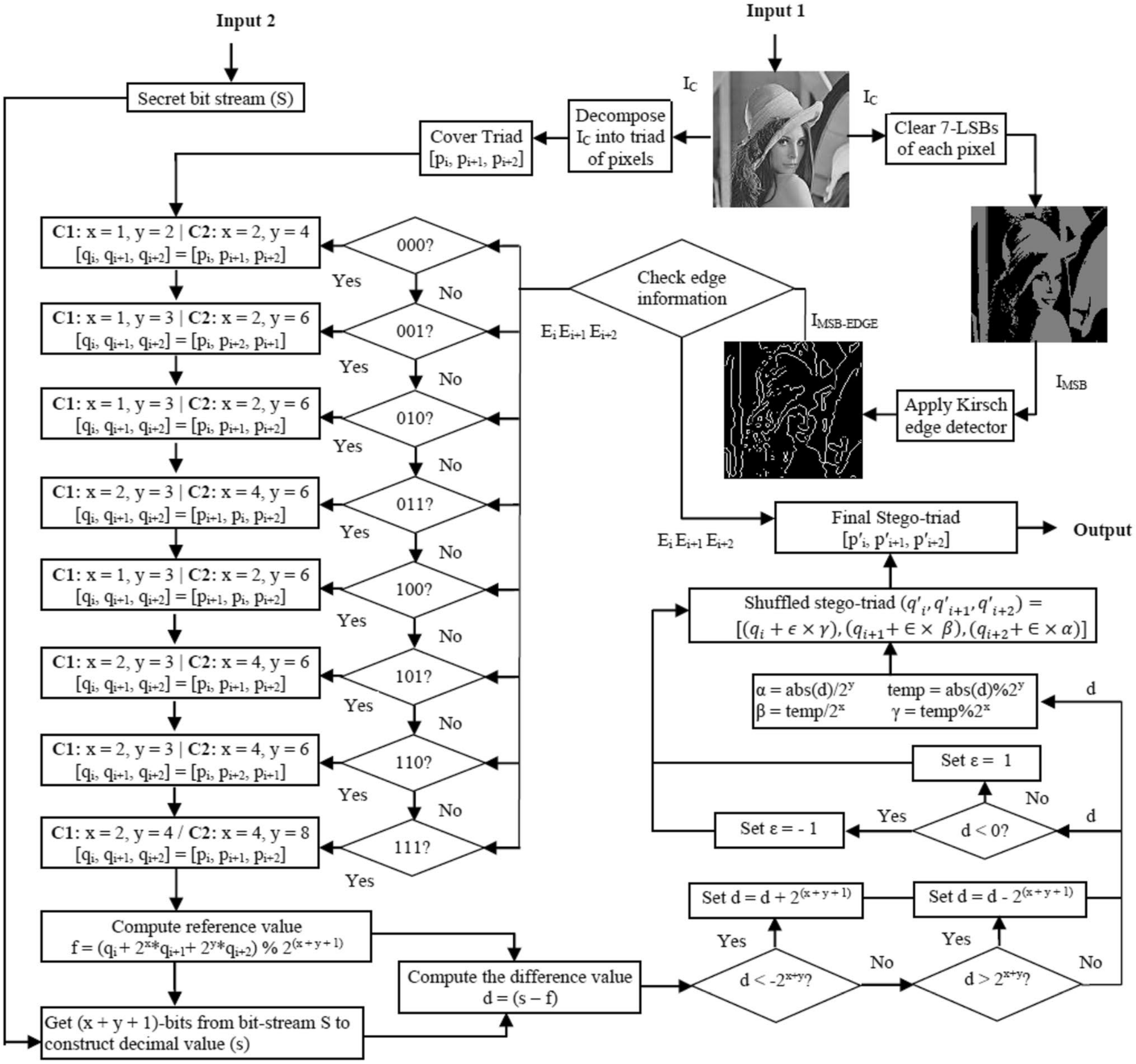

Input 2
Secret bit stream (S)
Input 1
Ic
Ic
Cover Triad [pi, pi+1, pi+2]
Decompose Ic into triad of pixels
Clear 7-LSBs of each pixel
C1: x = 1, y = 2 | C2: x = 2, y = 4 [qi, qi+1, qi+2] = [pi, pi+1, pi+2]
000?
Yes
No
Check edge information
C1: x = 1, y = 3 | C2: x = 2, y = 6 [qi, qi+1, qi+2] = [pi, pi+2, pi+1]
001?
Ei Ei+1 Ei+2
IMSB-EDGE
C1: x = 1, y = 3 | C2: x = 2, y = 6 [qi, qi+1, qi+2] = [pi, pi+1, pi+2]
010?
IMSB
Apply Kirsch edge detector
C1: x = 2, y = 3 | C2: x = 4, y = 6 [qi, qi+1, qi+2] = [pi+1, pi, pi+2]
011?
Final Stego-triad [p'i, p'i+1, p'i+2]
Output
Ei Ei+1 Ei+2
C1: x = 1, y = 3 | C2: x = 2, y = 6 [qi, qi+1, qi+2] = [pi+1, pi, pi+2]
100?
Shuffled stego-triad (q'i, q'i+1, q'i+2) = [(qi + ϵ × γ), (qi+1 + ∈ × β), (qi+2 + ∈ × α)]
C1: x = 2, y = 3 | C2: x = 4, y = 6 [qi, qi+1, qi+2] = [pi, pi+1, pi+2]
101?
α = abs(d)/2^y temp = abs(d)%2^y
β = temp/2^x γ = temp%2^x
d
C1: x = 2, y = 3 | C2: x = 4, y = 6 [qi, qi+1, qi+2] = [pi, pi+2, pi+1]
110?
Set ε = 1
Yes
No
d
C1: x = 2, y = 4 / C2: x = 4, y = 8 [qi, qi+1, qi+2] = [pi, pi+1, pi+2]
111?
Set ε = - 1
d < 0?
Set d = d + 2^(x+y+1)
Set d = d - 2^(x+y+1)
Compute reference value f = (qi + 2^x*qi+1 + 2^y*qi+2) % 2^(x+y+1)
Compute the difference value d = (s – f)
d < -2^(x+y)?
d > 2^(x+y)?
Yes
No
Get (x + y + 1)-bits from bit-stream S to construct decimal value (s)

Fig. 2 Flow diagram of the embedding phase used in the proposed steganography scheme

to the Kirsch edge detector. It is seen from Table 2 that the number of edges for Canny and LoG ranges from 742 to 2523 and 649 to 1965, respectively. However, at the same time, Kirsch edge detector offers the number of edge pixels in the range [1871–8588]. This very property of Kirsch edge detector-based embedding method enables an average payload enhancement of 0.2112 bpp and 0.4223 bpp than Canny edge detector and 0.232 bpp and 0.464 bpp than LoG edge detector, respectively. In this regard, an average PSNR drop of 2 ~ 3 dB has been compensated. However, the resulting average PSNR for both cases falls above the acceptable level. Similarly, the SSIM values are summarized for all three edge detection-based steganographic methods from which the average SSIM values are computed. It is also observed from Tables 3 and 4 that the proposed method offers uniform SSIM for all 10 images and as the payload increases, corresponding SSIM value drops but never falls beyond the acceptable range [0.90–0.99].

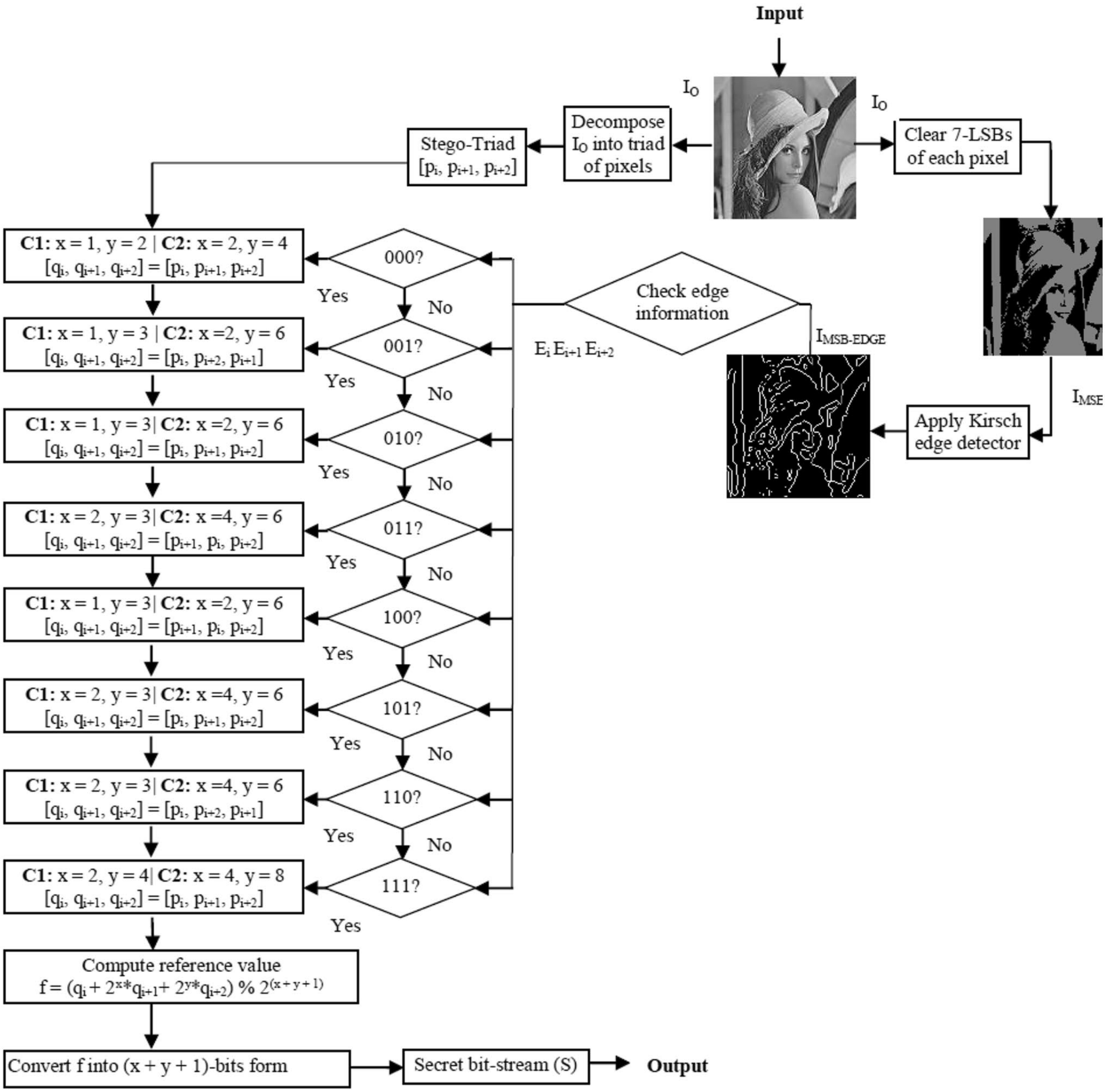
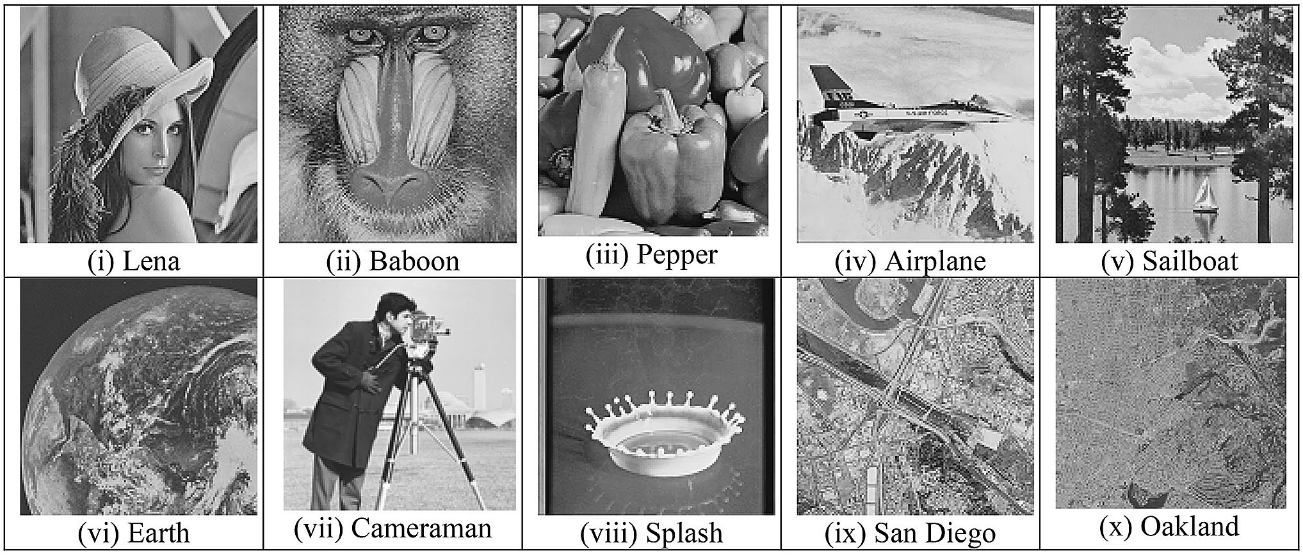

Input
I_O
I_O
Stego-Triad [p_i, p_i+1, p_i+2]
Decompose I_O into triad of pixels
Clear 7-LSBs of each pixel
Check edge information
E_i E_i+1 E_i+2
I_MSB-EDGE
I_MSE
Apply Kirsch edge detector
C1: x = 1, y = 2 | C2: x = 2, y = 4
[q_i, q_i+1, q_i+2] = [p_i, p_i+1, p_i+2]
000?
Yes
No
C1: x = 1, y = 3 | C2: x =2, y = 6
[q_i, q_i+1, q_i+2] = [p_i, p_i+2, p_i+1]
001?
Yes
No
C1: x = 1, y = 3| C2: x =2, y = 6
[q_i, q_i+1, q_i+2] = [p_i, p_i+1, p_i+2]
010?
Yes
No
C1: x = 2, y = 3| C2: x =4, y = 6
[q_i, q_i+1, q_i+2] = [p_i+1, p_i, p_i+2]
011?
Yes
No
C1: x = 1, y = 3| C2: x =2, y = 6
[q_i, q_i+1, q_i+2] = [p_i+1, p_i, p_i+2]
100?
Yes
No
C1: x = 2, y = 3| C2: x =4, y = 6
[q_i, q_i+1, q_i+2] = [p_i, p_i+1, p_i+2]
101?
Yes
No
C1: x = 2, y = 3| C2: x =4, y = 6
[q_i, q_i+1, q_i+2] = [p_i, p_i+2, p_i+1]
110?
Yes
No
C1: x = 2, y = 4| C2: x = 4, y = 8
[q_i, q_i+1, q_i+2] = [p_i, p_i+1, p_i+2]
111?
Yes
Compute reference value
f = (q_i + 2^x*q_i+1 + 2^y*q_i+2) % 2^(x + y + 1)
Convert f into (x + y + 1)-bits form
Secret bit-stream (S)
Output

Fig. 3 Flow diagram of the extraction phase used in the proposed steganography scheme

Table 5 demonstrates the comparative analysis in terms of Payload (bpp) and PSNR (dB) among the proposed method and some state-of-the-art methods [8, 9, 7, 15, 17] for 'Lena', 'Baboon', 'Pepper', 'Airplane' and 'Sailboat', respectively. Considering the results of Case 1, one can ensure that both payload and PSNR have been improved in the proposed method compared to the state-of-the-art methods. For better understanding, one can see that the average payload improvement for the proposed method over existing methods that may not be too high but the average PSNR hikes are: 4.06 dB, 6.19 dB, 1.21 dB and 4.73 dB in compared to the methods reported in [7–9] and [15], respectively. These hikes are really impressive as far as the quality is concerned. Similarly, in comparison with Atta et al.'s method [17], it is a fact that the average payload of proposed method is lagging by 0.08 bpp (1310 bits only) but at the same time, the average PSNR has been enhanced by 1.27 dB. Hence, this comparative assessment ensures that the performance of the proposed scheme is better than the existing schemes in terms of average payload and average PSNR or both. Further, to extend the payload up to significant amount, the results obtained in



(i) Lena (ii) Baboon (iii) Pepper (iv) Airplane (v) Sailboat (vi) Earth (vii) Cameraman (viii) Splash (ix) San Diego (x) Oakland

Fig. 4 Different gray-scale cover images of dimension 129 × 129 used for the current experimentation

Table 2 Experimental results of the proposed scheme in Case 1 and Case 2 for the images of dimension 129 × 129

Cover image	No. of edges	Case 1 (C1)					Case 2 (C2)				
		Payload (bpp)	PSNR (dB)	SSIM	MSE	UQI	Payload (bpp)	PSNR (dB)	SSIM	MSE	UQI
Lena	4561	1.6074	46.62	0.9962	1.41	0.9911	2.8814	**37.35**	0.9676	**11.95**	0.9105
Baboon	**8588**	**1.8494**	44.90	0.9962	2.09	0.9962	**3.3654**	31.81	0.9386	42.78	0.9398
Pepper	4129	1.5814	46.92	0.9972	1.32	0.9954	2.8295	34.60	0.9698	22.50	0.9616
Airplane	3999	1.5736	46.96	0.9964	1.30	0.9815	2.8139	34.75	0.9719	21.73	0.9207
Sailboat	5015	1.6346	46.42	0.9978	1.47	0.9915	2.9360	33.97	**0.9794**	26.05	0.9583
Earth	6571	1.7282	45.68	0.9967	1.75	0.9843	3.1230	32.94	0.9547	33.02	0.9181
Cameraman	2838	1.5038	47.78	0.9963	1.08	**0.9999**	2.6744	36.11	0.9753	15.92	0.8496
San Diego	7698	1.7959	45.21	**0.9982**	1.95	0.9984	3.2585	32.28	0.9691	38.46	**0.9716**
Oakland	8121	1.8213	45.01	0.9952	2.05	0.9945	3.3093	32.23	0.9252	38.85	0.9173
Splash	1871	1.4457	**48.47**	0.9955	**0.92**	0.9837	2.5581	**37.35**	0.9676	**11.95**	0.9105
Avg. Case	**5339**	**1.6541**	**46.39**	**0.9965**	**1.53**	**0.9916**	**2.9749**	**34.33**	**0.9619**	**26.32**	**0.9258**

The bold values in last row of a table represents the average case result. However, the bold values in other rows of selected columns represent the maximum values obtained using the proposed method for the said column name

Case 2 can be compared with the state-of-the-art methods. In that case the payload enhancement of the proposed method is at least 70% more than all these existing methods, however PSNR value never crosses the threshold level.

To ensure the superiority of the proposed scheme, we have further compared the proposed method with the leading edge detection-based steganography schemes published in the last few years [16, 18, 19, 20]. The experiments have been conducted on the 'Baboon' image and the results of the same are shown in Table 6. Since the Kirsch operator in the proposed method offers a wider edge region, the proposed method results in 0.3963 bpp (i.e. 6493 bits) more payload at the cost of 1 ~ 1.5 dB of image quality distortion over existing methods [16, 18, 20]. However, the proposed scheme offers both payload and quality improvement (+ 0.44 bpp and + 1.76 dB) than the Ghosal et al.'s method [19]. More than 130% payload improvement has been achieved using the result of Case 2, however the quality has been compromised with the methods cited in [16, 18, 19, 20].

6 Security analysis

For testing the robustness of the proposed method, we perform a steganalysis test on the obtained stego-images in order to understand the impact of a possible external attack

Table 3 Experimental results of the proposed scheme (Case 1) using different edge detection methods for the images of dimension 129 × 129

Cover image	Canny				LoG				Kirsch			
	No. of edges	Payload (bpp)	PSNR (dB)	SSIM	No. of Edges	Payload (bpp)	PSNR (dB)	SSIM	No. of edges	Payload (bpp)	PSNR (dB)	SSIM
Lena	1800	1.4414	48.60	0.9973	1467	1.4214	48.87	0.9973	4561	1.6074	46.62	0.9962
Baboon	2507	1.4839	48.03	0.9980	1879	1.4462	48.49	0.9982	**8588**	**1.8494**	44.90	0.9962
Pepper	1604	1.4297	48.79	0.9976	1370	1.4156	48.84	0.9976	4129	1.5814	46.92	0.9972
Airplane	1474	1.4219	48.88	0.9967	1287	1.4106	49.09	0.9967	3999	1.5736	46.96	0.9964
Sailboat	1693	1.4350	48.72	0.9980	1513	1.4242	48.87	0.9980	5015	1.6346	46.42	0.9978
Earth	2321	1.4728	48.18	0.9977	1850	1.4445	48.49	0.9978	6571	1.7282	45.68	0.9967
Cameraman	1191	1.4049	49.10	0.9964	978	1.3921	49.35	0.9964	2838	1.5038	47.78	0.9963
San Diego	2394	1.4771	48.07	**0.9990**	1825	1.4430	48.58	**0.9990**	7698	1.7959	45.21	**0.9982**
Oakland	**2523**	**1.4849**	47.96	0.9972	**1965**	**1.4514**	48.44	0.9974	8121	1.8213	45.01	0.9952
Splash	742	1.3779	**49.64**	0.9957	649	1.3723	**49.69**	0.9957	1871	1.4457	**48.47**	0.9955
Avg. Case	**1824**	**1.4429**	**48.59**	**0.9973**	**1478**	**1.4221**	**48.87**	**0.9974**	**5339**	**1.6541**	**46.39**	**0.9965**

The bold values in last row of a table represents the average case result. However, the bold values in other rows of selected columns represent the maximum values obtained using the proposed method for the said column name

Table 4 Experimental results of the proposed scheme (Case2) using different edge detection methods for the images of dimension 129 × 129

Cover images	Canny				LoG				Kirsch			
	No. of edges	Payload (bpp)	PSNR (dB)	SSIM	No. of Edges	Payload (bpp)	PSNR (dB)	SSIM	No. of edges	Payload (bpp)	PSNR (dB)	SSIM
Lena	1800	2.5496	37.58	0.9748	1467	2.5096	38.12	0.9753	4561	2.8814	**37.35**	**0.9676**
Baboon	2507	2.6346	36.45	0.9749	1879	2.5591	37.29	0.9793	**8588**	**3.3654**	31.81	0.9386
Pepper	1604	2.5261	37.89	0.9814	1370	2.4979	38.39	0.9818	4129	2.8295	34.60	0.9698
Airplane	1474	2.5104	38.17	0.9779	1287	2.4880	38.43	0.9778	3999	2.8139	34.75	0.9719
Sailboat	1693	2.5368	37.72	0.9861	1513	2.5151	37.95	0.9854	5015	2.9360	33.97	0.9794
Earth	2321	2.6122	36.78	0.9768	1850	2.5556	37.37	0.9789	6571	3.1230	32.94	0.9547
Cameraman	1191	2.4764	38.70	0.9776	978	2.4508	39.24	0.9775	2838	2.6744	36.11	0.9753
San Diego	2394	2.6210	36.45	**0.9869**	1825	2.5526	37.49	**0.9893**	7698	3.2585	32.28	0.9691
Oakland	**2523**	**2.6365**	36.41	0.9670	**1965**	**2.5694**	37.18	0.9713	8121	3.3093	32.23	0.9252
Splash	742	2.4225	**39.86**	0.9713	649	2.4113	**40.04**	0.9725	1871	2.5581	**37.35**	**0.9676**
Avg. Case	**1824**	**2.5526**	**37.60**	**0.9774**	**1478**	**2.5109**	**38.15**	**0.9789**	**5339**	**2.9749**	**34.33**	**0.9619**

The bold values in last row of a table represents the average case result. However, the bold values in other rows of selected columns represent the maximum values obtained using the proposed method for the said column name

Table 5 Performance comparison of the proposed method compared to some state-of-the-art methods in terms of payload and PSNR

Image	Methods	Payload (bpp)	PSNR (dB)
Lena	Shen et al. [8]	1.54	42.46
	Shen et al. [9]	1.68	40.96
	Lee et al. [7]	1.75	44.31
	Tseng et al.'s scheme [15]	1.66	41.03
	Atta et al. [17]	1.85	43.99
	Proposed scheme	**1.73**	**45.59**
		2.88	**37.35**
Baboon	Shen et al. [8]	1.69	38.88
	Shen et al. [9]	1.90	36.12
	Lee et al. [7]	1.73	44.28
	Tseng et al.'s scheme [15]	1.80	40.22
	Atta et al. [17]	1.84	44.43
	Proposed scheme	**1.97**	**44.12**
		3.36	**31.81**
Pepper	Shen et al. [8]	1.53	42.68
	Shen et al. [9]	1.68	39.99
	Lee et al. [7]	1.73	44.32
	Tseng et al.'s scheme [15]	1.65	40.88
	Atta et al. [17]	1.82	44.48
	Proposed scheme	1.69	45.95
		2.82	**34.60**
Airplane	Shen et al. [8]	**1.54**	**42.17**
	Shen et al. [9]	1.70	40.10
	Lee et al. [7]	1.68	44.45
	Tseng et al.'s scheme [15]	1.65	41.02
	Atta et al. [17]	1.83	44.15
	Proposed scheme	**1.63**	**46.41**
		2.81	**34.75**
Sailboat	Shen et al. [8]	1.56	41.29
	Shen et al. [9]	1.72	39.65
	Lee et al. [7]	–	–
	Tseng et al.'s scheme [15]	1.67	40.95
	Atta et al. [17]	1.85	44.36
	Proposed scheme	**1.73**	**45.72**
		2.93	**33.97**
Average case	Shen et al. [8]	1.57	41.49
	Shen et al. [9]	1.73	39.36
	Lee et al. [7]	1.72	44.34
	Tseng et al.'s scheme [15]	1.68	40.82
	Atta et al. [17]	1.83	44.28
	Proposed scheme	***1.75***	***45.55***
		2.96	***34.49***

The bold values in last row of a table represents the average case result

Table 6 Comparison of PSNR (dB) values of the proposed method and existing edge detection-based methods on 'Baboon' image

Methods	Payload (bpp)	PSNR (dB)
Bai et al.'s method [16] (Canny)	1.4531	44.6229
Gaurav and Ghanekar's method [18]	1.4531	45.4115
Ghosal et al.'s method [19]	1.4100	43.1400
Setiadi's method [20]	1.4531	46.4381
Proposed scheme	**1.8494**	**44.9000**
	3.3654	**31.8100**

The bold values in last row of a table represents the average case result

on the system. The method as illustrated by the StegExpose tool [24], performs a fast fusion of standard, well-known steganalysis techniques in order to develop an optimized method. Proposed by Dumitrescu et al. [25], the analysis by the Primary Sets forms a quadratic equation which captures the statistical dissimilarities between the cardinality of neighbouring pixels drawn out of a consistently toned image. LSB embedding may lead to a change in the cardinality, which is the driving factor being Primary Set analysis to be successful. RS Analysis [26], on the other hand, works on the principle of lossless data embedding and works with three different pixel groups—Regular, Singular and Unusable. It follows a paradigm of flipping, defined as a permutation of gray layer consisting of 2-cycles and finds stego-images by estimating the intersection point between the percentage of flipped LSB's and relative number of regular and singular groups.

To perform a fast fusion and save computation time, StegExpose iterates over a number of steps and works towards only analyzing seemingly possible stego files. At each step, a detector component is added and the cumulative arithmetic mean gets updated. If this mean falls below a certain threshold, then the image is considered to be tampered. The results in Table 7 show that out of the 10 images of two different cases each, only 1 image is above the stego threshold. This goes onto show that for 95% of the stego-images, the steganalysis does not detect the presence of external embedding and passes the test.

7 Conclusion

In this paper, a novel steganography method is proposedusing the concept of the Kirsch edge detector. One of the notable advantages of the Kirsch edge detector is the ability of producing more edge pixels and it ensures that the variable payload can be achieved by this method. The PSNR values obtained by the proposed method are always above the acceptable level (i.e. 30 dB). Performance of in terms of

Table 7 Security analysis of the proposed steganography method for the 10 benchmark images

Name of stego-image file	Case	Above stego threshold?	Secret message size (in byte)	Primary sets	RS analysis	Fusion (mean)
lena129_stego_kirsch.bmp	C1	FALSE	572	0.13908	0.050459	0.09477
	C2	FALSE	220	0.033294	0.039547	0.036421
baboon129_stego_kirsch.bmp	C1	TRUE	1288	0.213069	0.213819	0.213444
	C2	FALSE	779	NaN	0.129059	0.129059
pepper129_stego_kirsch.bmp	C1	FALSE	216	0.012527	0.058934	0.03573
	C2	FALSE	615	0.122811	0.081105	0.101958
airplane129_stego_kirsch.bmp	C1	FALSE	496	0.106255	0.058065	0.08216
	C2	FALSE	300	0.038727	0.060756	0.049742
sailboat129_stego_kirsch.bmp	C1	FALSE	421	0.06281	0.076559	0.069685
	C2	FALSE	646	0.07902	0.135164	0.107092
earth129_stego_kirsch.bmp	C1	FALSE	416	0.123947	0.01377	0.068858
	C2	FALSE	183	0.003755	0.056878	0.030317
cameraman129_stego_kirsch.bmp	C1	FALSE	292	0.035592	0.061243	0.048418
	C2	FALSE	430	0.086026	0.056592	0.071309
san_diego129_stego_kirsch.bmp	C1	FALSE	679	NaN	0.112453	0.112453
	C2	FALSE	473	NaN	0.078313	0.078313
splash129_stego_kirsch.bmp	C1	FALSE	147	0.032344	0.016498	0.024421
	C2	FALSE	192	0.012011	0.051577	0.031794
oakland129_stego_kirsch.bmp	C1	FALSE	606	0.126371	0.074443	0.100407
	C2	FALSE	327	0.035111	0.07334	0.054225

other metrics such as MSE, SSIM and UQI also proves the effectiveness of the proposed scheme. Further, the average PSNR values of the proposed method are better than both conventional and edge-based steganography methods and at the same time it maintains a high payload. Another significant advantage of proposed scheme is the ability to avoid the storing of extra edge information while the actual embedding takes place. The robustness of the stego-images has been tested using the StegExpose tool and it is found from the test cases that for 95% of the stego-images, the steganalysis does not detect the presence of external embedding.

In future, the payload of the proposed scheme can be improved by dilating the edge images prior to embedding so that a huge number of edge pixels can be obtained, which eventually would help in increasing the payload. The transform domain methods may also be investigated to ensure high robustness and imperceptibility. Some pseudo-random sequences may also be incorporated for ensuring better security of the proposed scheme.

References

1. Yan, C., Shao, B., Yuxuan, W., Yue, G.: Deep multi-view enhancement hashing for image retrieval. IEEE Trans Pattern Anal Mach Intell (2020). https://doi.org/10.1109/TPAMI.2020.2975798
2. Yan, C., Shao, B., Zhao, H., Ning, R., Zhang, Y., Feng, X.: 3D room layout estimation from a single RGB image. IEEE Trans Multimed (2020). https://doi.org/10.1109/TMM.2020.2967645
3. Yan, C., Li, Z., Zhang, Y., Liu, Y., Ji, X., Zhang,Y.: Depth image denoising using nuclear norm and learning graph model. ACM Trans. Multimed. Comput. Commun. Appl. (2020). https://arxiv.org/pdf/2008.03741.pdf
4. Bender, W., Gruhl, D., Morimoto, N., Lu, A.: Techniques for data hiding. IBM Syst J **35**(3–4), 313–336 (1996)
5. Wu, D.-C., Tsai, W.-H.: A Steganographic method for images by pixel-value differencing. Pattern Recogn Lett. **24**(9–10), 1613–1626 (2003)
6. Zhang, X., Wang, S.: Efficient steganographic embedding by exploiting modification direction. IEEE Commun. Lett. **10**(11), 1–3 (2006)
7. Lee C. F., Wang Y. R., Chang C. C.: A Steganographic method with high embedding capacity by improving exploiting modification direction. Proceedings of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kaohsiung, Vol. 1, pp. 497–500 (2007).
8. Shen, S.-Y., Huang, L.-H.: A data hiding scheme using pixel value differencing and improving exploiting modification directions. Comput. Secur. **48**, 131–141 (2015)
9. Shen, S.-Y., Huang, L.-H., Wu, S.-S.: A novel adaptive data hiding based on improved EMD and interpolation. Multimed. Tools Appl. (2017). https://doi.org/10.1007/s11042-017-4905-5,pp.1-17
10. Mukherjee, S., Sanyal, G.: A multi level image steganography methodology based on adaptive PMS and block based pixel swapping. Multimed. Tools Appl. **78**, 17607–17622 (2019). https://doi.org/10.1007/s11042-018-7127-6
11. Abdulla, A.A., Sellahewa, H., Jassim, S.A.: Improving embedding efficiency for digital steganography by exploiting similarities

between secret and cover images. Multimed. Tools Appl. **78**, 17799–17823 (2019). https://doi.org/10.1007/s11042-019-7166-7

12. Chatterjee, A., Ghosal, S.K., Sarkar, R.: LSB based steganography with OCR: an intelligent amalgamation. Multimed. Tools Appl. **79**, 11747–11765 (2020). https://doi.org/10.1007/s11042-019-08472-6
13. Mohamed, N., Baziyad, M., Rabie, T., et al.: L*a*b* color space high capacity steganography utilizing quad-trees. Multimed Tools Appl **79**, 25089–25113 (2020). https://doi.org/10.1007/s11042-020-09129-5
14. Chen, W.J., Chang, C.C., Le, T.H.N.: High payload Steganography mechanism using hybrid edge detector. Expert Syst. Appl. **37**, 3292–3301 (2010)
15. Tseng, H.W., Leng, H.S.: High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion. IET Image Process. **8**, 647–654 (2014)
16. Bai, J., Chang, C.-C., Nguyen, T.-S., Zhu, C., Liu, Y.: A high payload steganographic algorithm based on edge detection. Displays **46**, 42–51 (2017). https://doi.org/10.1016/j.displa.2016.12.004
17. Atta, R., Ghanbari, M.: A high payload steganography mechanism based on wavelet packet transformation and neutrosophic set, journal of visual communication & image representation. Elsevier (2018). https://doi.org/10.1016/j.jvcir.2018.03.009
18. Gaurav, K., Ghanekar, U.: Image steganography based on Canny edge detection, dilation operator and hybrid coding. J. Inf. Secur. Appl. **41**, 41–51 (2018). https://doi.org/10.1016/j.jisa.2018.05.001
19. Ghosal, S.K., Mandal, J.K., Sarkar, R.: High payload image steganography based on Laplacian of Gaussian (LoG) edge detector. Multimed. Tools Appl. **77**, 30403–30418 (2018). https://doi.org/10.1007/s11042-018-6126-y
20. Setiadi, D.R.I.M.: Improved payload capacity in LSB image steganography uses dilated hybrid edge detection. J. King Saud Univ Comput Inf Sci (2019). https://doi.org/10.1016/j.jksuci.2019.12.007
21. Kumar, S., Singh, A., Kumar, M.: Information hiding with adaptive steganography based on novel fuzzy edge identification. Defence Technol. **15**(2), 162–169 (2019). https://doi.org/10.1016/j.dt.2018.08.003
22. Kirsch, R.: Computer determination of the constituent structure of biological images. Comput. Biomed. Res. **4**, 315–328 (1971)
23. Weber A. G., The USC-SIPI Image Database: Version 5, Original release: October 1997, Signal and Image Processing Institute, University of Southern California, Department of Electrical Engineering. https://sipi.usc.edu/database/. Accessed 11 Jul 2019.
24. Boehm, B.: Stegexpose-A tool for detecting LSB steganography. arXiv preprint arXiv:1410.6656 (2014).
25. Dumitrescu, S., Xiaolin, W., Nasir M.: On steganalysis of random LSB embedding in continuous-tone images. Proceedings. International Conference on Image Processing. Vol. 3. IEEE (2002).
26. Fridrich, J., Goljan, M., Dui, R.: Reliable detection of LSB steganography in color and grayscale images. Proceedings of the ACM Workshop on Multimedia and Security, pp. 27–30 (2001).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Terms and Conditions