Signal Processing 90 (2010) 727–752

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Review

Digital image steganography: Survey and analysis of current methods

Abbas Cheddad *, Joan Condell, Kevin Curran, Paul Mc Kevitt

School of Computing and Intelligent Systems, Faculty of Computing and Engineering, University of Ulster at Magee, Londonderry, BT48 7JL, Northern Ireland, UK

ARTICLE INFO

Article history:
Received 1 December 2008
Received in revised form
17 August 2009
Accepted 18 August 2009
Available online 6 September 2009

Keywords:
Digital image steganography
Spatial domain
Frequency domain
Adaptive steganography
Security

ABSTRACT

Steganography is the science that involves communicating secret data in an appropriate multimedia carrier, e.g., image, audio, and video files. It comes under the assumption that if the feature is visible, the point of attack is evident, thus the goal here is always to conceal the very existence of the embedded data. Steganography has various useful applications. However, like any other science it can be used for ill intentions. It has been propelled to the forefront of current security techniques by the remarkable growth in computational power, the increase in security awareness by, e.g., individuals, groups, agencies, government and through intellectual pursuit. Steganography's ultimate objectives, which are undetectability, robustness (resistance to various image processing methods and compression) and capacity of the hidden data, are the main factors that separate it from related techniques such as watermarking and cryptography. This paper provides a state-of-the-art review and analysis of the different existing methods of steganography along with some common standards and guidelines drawn from the literature. This paper concludes with some recommendations and advocates for the object-oriented embedding mechanism. Steganalysis, which is the science of attacking steganography, is not the focus of this survey but nonetheless will be briefly discussed.

Contents

* Corresponding author.
E-mail addresses: cheddad-a@email.ulster.ac.uk, cheddad@gmail.com (A. Cheddad).

0165-1684/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2009.08.010

1. Introduction

The standard and concept of "What You See Is What You Get (WYSIWYG)" which we encounter sometimes while printing images or other materials, is no longer precise and would not fool a steganographer as it does not always hold true. Images can be more than what we see with our Human Visual System (HVS); hence, they can convey more than merely 1000 words.

For decades people strove to develop innovative methods for secret communication. The remainder of this introduction highlights briefly some historical facts and attacks on methods (also known as steganalysis). A thorough history of steganography can be found in the literature [1–3].

Three techniques are interlinked, steganography, watermarking and cryptography. The first two are quite difficult to tease apart especially for those coming from different disciplines. Fig. 1 and Table 1 may eradicate such confusion. The work presented here revolves around steganography in digital images and does not discuss other types of steganography (such as linguistic or audio).

1.1. Nomenclature

Intuitively, this work makes use of some terms commonly used by steganography and watermarking communities. The term "cover image" will be used throughout this paper to describe the image designated to carry the embedded bits. We will be referring to an

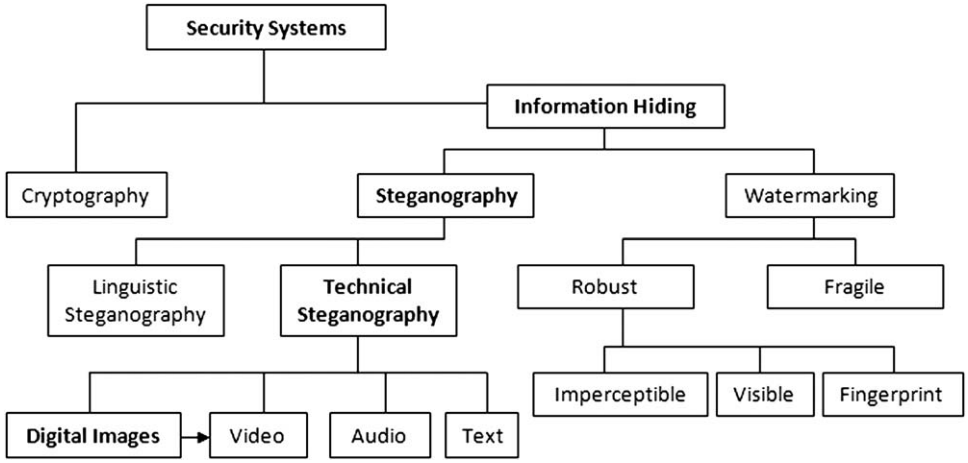

Fig. 1. The different embodiment disciplines of information hiding. The arrow indicates an extension and bold face indicates the focus of this study.

Table 1
Comparison of steganography, watermarking and encryption.

Criterion/ method	Steganography	Watermarking	Encryption
Carrier	Any digital media	Mostly image/audio files	Usually text based, with some extensions to image files
Secret data	Payload	Watermark	Plain text
Key	Optional		Necessary
Input files	At least two unless in self-embedding		One
Detection	Blind	Usually informative (i.e., original cover or watermark is needed for recovery)	Blind
Authentication	Full retrieval of data	Usually achieved by cross correlation	Full retrieval of data
Objective	Secrete communication	Copyright preserving	Data protection
Result	Stego-file	Watermarked-file	Cipher-text
Concern	Delectability/ capacity	Robustness	Robustness
Type of attacks	Steganalysis	Image processing	Cryptanalysis
Visibility	Never	Sometimes (see Fig. 2)	Always
Fails when	It is detected	It is removed/replaced	De-ciphered
Relation to cover	Not necessarily related to the cover. The message is more important than the cover	Usually becomes an attribute of the cover image. The cover is more important than the message	N/A
Flexibility	Free to choose any suitable cover	Cover choice is restricted	N/A
History	Very ancient except its digital version	Modern era	Modern era

image with embedded data, called herein payload, as "stego-image". Further "steganalysis" or "attacks" refer to different image processing and statistical analysis approaches that aim to break or attack steganography algorithms (Fig. 2).

1.2. Ancient steganography

The word steganography is originally derived from Greek words which mean "Covered Writing". It has been used in various forms for thousands of years. In the 5th century BC Histaiacus shaved a slave's head, tattooed a message on his skull and the slave was dispatched with the message after his hair grew back [1–4]. In Saudi Arabia at the King Abdulaziz City of science and technology, a project was initiated to translate into English some ancient Arabic manuscripts on secret writing which are believed to have been written 1200 years ago. Some of these manuscripts were found in Turkey and Germany [5]. Five hundred years ago, the Italian mathematician Jérôme Cardan reinvented a Chinese ancient method of secret writing. The scenario goes as follows: a paper mask with holes is shared among two parties, this mask is placed over a blank paper and the sender writes his secret message through the holes then takes the mask off and fills the blanks so that the message appears as an innocuous text as shown in Fig. 3. This method is credited to Cardan and is called Cardan Grille [4].

It was also reported that the Nazis invented several steganographic methods during World War II such as Microdots, and have reused invisible ink and null ciphers. As an example of the latter a message was sent by a Nazi spy that read: "Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils." Using the 2nd letter from each word the secret message reveals: "Pershing sails from NY June 1" [2,6,7].

In 1945, Morse code was concealed in a drawing (see Fig. 4). The hidden information is encoded onto the stretch of grass alongside the river. The long grass denoted a line and the short grass denoted a point. The decoded message read: "Compliments of CPSA MA to our chief Col Harold R. Shaw on his visit to San Antonio May 11th 1945" [8].

1.3. The digital era of steganography

With the boost in computer power, the internet and with the development of digital signal processing (DSP), information theory and coding theory, steganography has gone "digital". In the realm of this digital world steganography has created an atmosphere of corporate vigilance that has spawned various interesting applications, thus its continuing evolution is guaranteed. Contemporary information hiding is due to [9]. One of the earliest methods to discuss digital steganography is credited to Kurak and McHugh [10], who proposed a method which resembles embedding into the 4 LSBs (least significant bits). They examined image downgrading and contamination which is known now as image-based steganography.

Cyber-crime is believed to benefit from this digital revolution. Hence an immediate concern was shown on the possible use of steganography by terrorists following a

Fig. 2. Media TV channels usually have their logos watermark for their broadcasting.

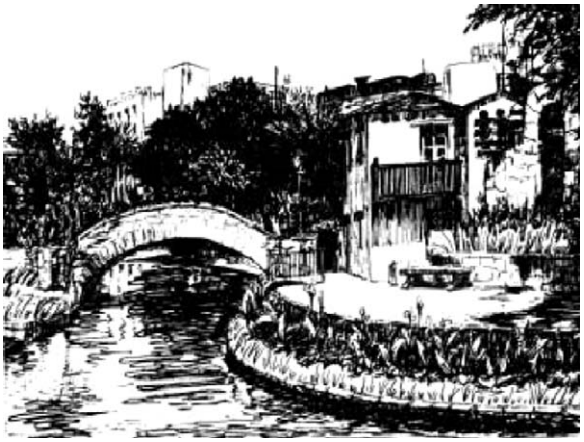

Fig. 4. Concealment of Morse code (1945). The hidden information is encoded onto the grass length alongside the river [8].

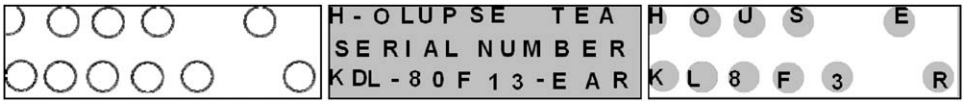

Fig. 3. Cardan Grille: an illustration, keeping in mind that the Grill has no fixed pattern: (left) the mask, (middle) the cover and (right) the secret message revealed.

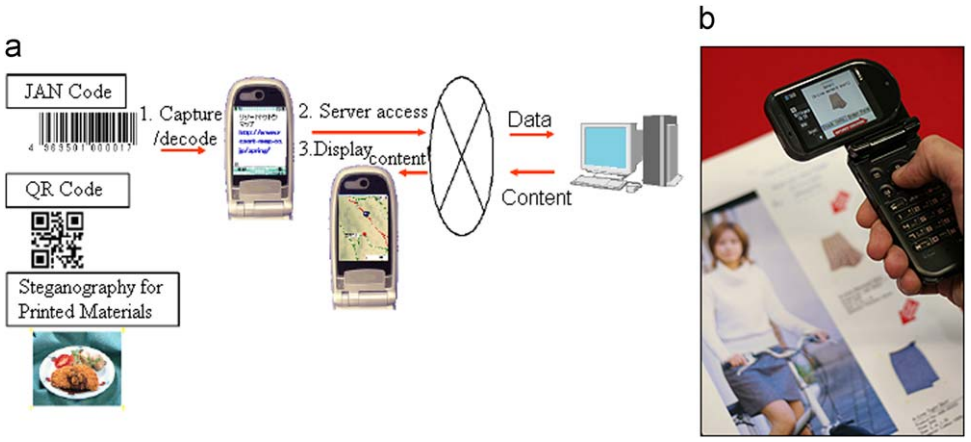

Fig. 5. Fujitsu exploitation of steganography: (a) a sketch representing the concept and (b) the idea deployed into a mobile phone shown at an exhibition recently.[3]

report in USA TODAY.[1] Cyber-planning or the "digital menace" as Lieutenant Colonel Timothy L. Thomas defined it, is difficult to control [11]. Provos and Honeyman [3], at the University of Michigan, scrutinized three million images from popular websites looking for any trace of steganography. They have not found a single hidden message. Despite the fact that they attributed several reasons to this failure it should be noted that steganography does not exist merely in still images. Embedding hidden messages in video and audio files is also possible. Examples exist in [12] for hiding data in music files, and even in a simpler form such as in Hyper Text Mark up Language (HTML), executable files (.EXE) and Extensible Markup Language (XML) [13]. This shows that USA TODAY's claim is not supported by a strong evidence, especially knowing that the writer of the above report resigned about two years later after editors determined that he had deceived them during the course of their investigation.[2]

This paper's focus is on the review of steganography in digital images. For a detailed survey on steganographic tools in other media from a forensic investigator's perspective the reader is referred to [14].

Section 2 briefly discusses the applications of steganography. Methods available in the literature are described in Section 3. The main discussions and comparisons focus on spatial domain methods, frequency domain methods and also adaptive methods in digital images. It will be shown that most of the steganographic algorithms discussed have been detected by steganalysis algorithms and thus a more robust approach needs to be developed and investigated. Section 4 will give a brief analysis and set it in context. Section 5 will discuss in brief the counterfeiting of steganography, a science known as steganalysis. A conclusion is provided in Section 6.

2. Steganography applications

Steganography is employed in various useful applications, e.g., copyright control of materials, enhancing robustness of image search engines and smart IDs (identity cards) where individuals' details are embedded in their photographs. Other applications are video–audio synchronization, companies' safe circulation of secret data, TV broadcasting, TCP/IP packets (for instance a unique ID can be embedded into an image to analyze the network traffic of particular users) [1], and also checksum embedding [15]. Petitcolas [16] demonstrated some contemporary applications, one of which was in Medical Imaging Systems where a separation is considered necessary for confidentiality between patients' image data or DNA sequences and their captions, e.g., physician, patient's name, address and other particulars. A link however, must be maintained between the two. Thus, embedding the patient's information in the image could be a useful safety measure and helps in solving such problems. Steganography would provide an ultimate guarantee of authentication that no other security tool may ensure. Miaou et al. [17] present an LSB embedding technique for electronic patient records based on bi-polar multiple-base data hiding. A pixel value difference between an original image and its JPEG version is taken to be a number conversion base. Nirinjan and Anand [18] and Li et al. [19] also discuss patient data concealment in digital images.

Inspired by the notion that steganography can be embedded as part of the normal printing process, the Japanese firm Fujitsu[3] is developing technology to encode data into a printed picture that is invisible to the human eye (data), but can be decoded by a mobile phone with a camera as exemplified in Fig. 5a and shown in action in Fig. 5b. The process takes less than one second as the embedded data is merely 12 bytes. Hence, users will be able to use their cellular phones to capture encoded data.

[1] USA TODAY: "Researchers: No secret bin Laden messages on sites". [Online]: 〈http://www.usatoday.com/tech/news/2001/10/17/bin-laden-site.htm#more〉.

[2] Jack Kelley's resignation: 〈www.usatoday.com/news/2004-01-16-reporter_x.htm〉.

[3] BBC News: Hiding messages in plain sight, available from: 〈http://news.bbc.co.uk/go/pr/fr/-/1/hi/technology/6361891.stm〉.

They charge a small fee for the use of their decoding software which sits on the firm's own servers. The basic idea is to transform the image colour scheme prior to printing to its hue, saturation and value components (HSV), then embed into the Hue domain to which human eyes are not sensitive. Mobile cameras can see the coded data and retrieve it. This application can be used for "doctor's prescriptions, food wrappers, billboards, business cards and printed media such as magazines and pamphlets" [20], or to replace barcodes.

The confidence in the integrity of visual imagery has been ruined by contemporary digital technology [21]. This led to further research pertaining to digital document forensics. As an example, Cheddad et al. [22] proposed a security scheme which protects scanned documents from forgery using self-embedding techniques. The method not

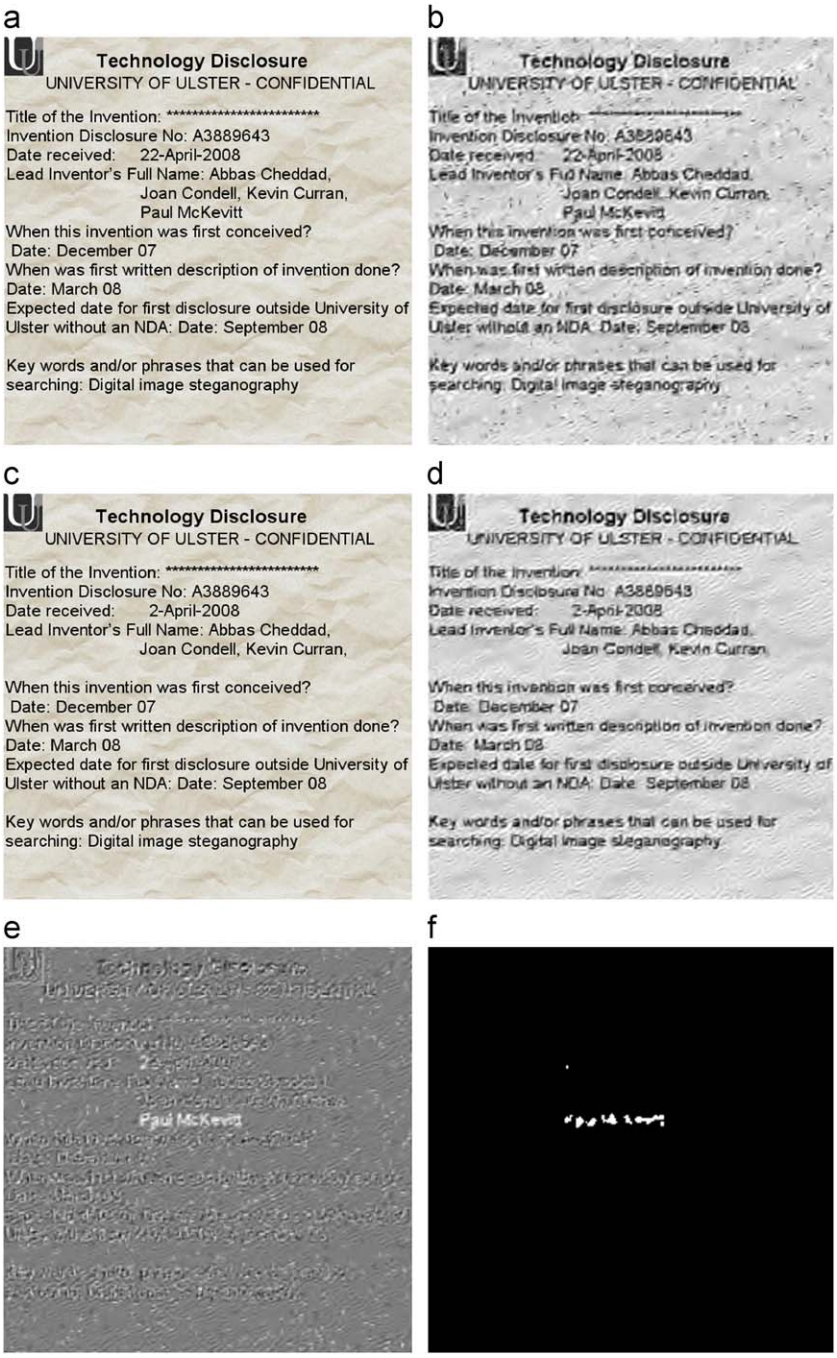

Fig. 6. Digital document forgery detection (a) Stego-image carrying self-duplicate (b), (c) attacked Stego-image, i.e., date received has changed and the 4th lead inventor's name has been removed, (d) inverse halftoning of the reconstructed hidden data from the attacked version, (e) error signal of (b) and (d), (f) after applying thresholding operation [22].

only points out forgery but also allows legal or forensics experts to gain access to the original document despite being manipulated (as can be seen from Fig. 6).

3. Steganography methods

This section attempts to give an overview of the most important steganographic techniques in digital images. The most popular image formats on the internet are graphics interchange format (GIF), Joint Photographic Experts Group (JPEG), and to a lesser extent—the portable network graphics (PNG). Most of the techniques developed were set up to exploit the structures of these formats with some exceptions in the literature that use the bitmap format (BMP) for its simple data structure.

We define the process of embedding as follows (a graphical representation is shown in Fig. 7):

Let C denote the cover carrier, i.e., image A, and C' the stego-image. Let K represent an optional key (a seed used to encrypt the message or to generate a pseudorandom noise which can be set to $\{\emptyset\}$ for simplicity) and let M be the message we want to communicate, i.e., image *B. Em* is an acronym for embedding and *Ex* for Extraction. Therefore:

$$\mathrm{Em} : C \oplus K \oplus M \rightarrow C' \tag{1}$$

$$\therefore \mathrm{Ex}(\mathrm{Em}(c,k,m)) \approx m, \forall c \in C,\ \ k \in K,\ \ m \in M \tag{2}$$

We will first discuss briefly some methods which exploit image formats. Then we will examine some of the dominant techniques bearing in mind that the most popular survey available on steganographic techniques was published ten years ago [23]. An evaluation of different spatial steganographic techniques applied especially to GIF images is also available [24].

In reference to the survey of Johnson et al. [23]:

- This paper is purely dedicated to steganography in image files (the most widespread research area) unlike in Johnson et al. who discuss in: Section 3.2.8 (Unused or reserved space in computer systems), Section 3.3.2 (Hiding information in digital sound), Section 3.3.3 (Echo hiding), Section 3.6.1 (Encoding information in formatted text), Section 3.7.1 (Mimics functions), Section 3.7.2 (Automated generation of English texts).
- Since the publication of Johnson et al. work, steganography has evolved dramatically. Therefore, an up-to-date survey was deemed necessary. In Johnson et al. work, the latest cited paper was published in 1999, which means their survey is now 10 years old.
- This paper's recommendations and method analysis can distinguish this initiative from that of Johnson et al. [23].
- The survey of Johnson et al. [23] appeared in the "Information hiding" book, which limits its distribution (i.e., cost matters especially for young researchers) compared to a Journal paper which can be more affordable.
- The classification, herein, of the techniques and that of Johnson et al. are different. Johnson et al. classify steganography techniques into: Substitution systems, transform domain techniques, spread spectrum techniques, statistical methods, distortion techniques, and cover generation methods.
- Johnson et al.'s survey neither talks about the history of steganography nor its applications (unlike this survey).
- Johnson et al.'s work has not included test images that can allow readers visualize the concepts.

In reference to the survey of Bailey and Curran [24]:

- The authors evaluate in their work some software that is applied in the spatial domain; mainly those supporting GIF formats (see Bailey and Curran [24, p. 62]). However, they did not discuss or evaluate the frequency domain software/methods and did not criticise the core algorithms.
- In Bailey and Curran's work, published three years ago, the latest cited paper was published in 2001. That means their survey, in fact, is 8 years old.
- They apply perceptual evaluation using a direct comparison between the original and stego-image files. Steganography assumes the unavailability of the original image.
- Their survey concludes the evaluation without recommendations or enhancements.

Section 3.2 discusses spatial domain techniques which generally uses a direct least significant bit (LSB) replacement technique. Section 3.3 discusses the frequency domain based methods such as discrete cosine transform (DCT), Fourier transform (FT) and discrete wavelet transform (DWT). Finally, the third sub-section will highlight the recent contribution in the domain which is termed perceptual masking (PM) or adaptive steganography (AS).

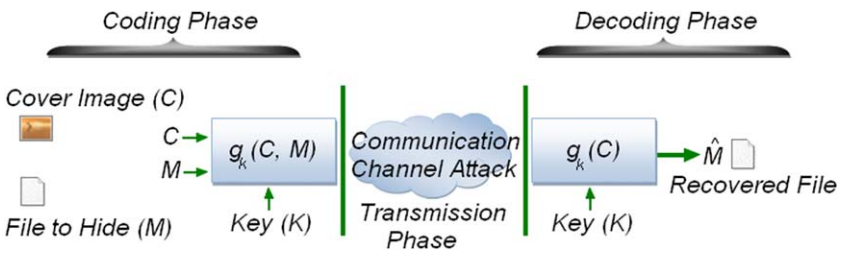

Fig. 7. Communication-theoretical view of a generic embedding process: C denotes cover image, M denotes the data to hide.

The categorization of steganographic algorithms into the three categories, namely, spatial domain, frequency domain and adaptive methods, is unique to this work and there is no claim that it is a standard categorization. Adaptive methods can either be applied in the spatial or frequency domains; as such they are regarded as special cases. We opt not to include image-format based steganography here as it is a naïve implementation and extremely prone to detection.

3.1. Steganography exploiting the image format

Steganography can be accomplished by simply feeding into a Windows OS command window, e.g., Windows XP) the following code: `C:> Copy Cover.jpg /b + Message.txt /b Stego.jpg`

What this code does is that it appends the secret message found in the text file "Message.txt" into the JPEG image file "Cover.jpg" and produces the stego-image "Stego.jpg". The idea behind this is to abuse the recognition of EOF (End of file). In other words, the message is packed and inserted after the EOF tag. When Stego.jpg is viewed using any photo editing application, the latter will just display the picture ignoring anything coming after the EOF tag. However, when opened in Notepad for example, our message reveals itself after displaying some data as shown in Fig. 8. The embedded message does not impair the image quality. Neither image histograms nor visual perception can detect any difference between the two images due to the secret message being hidden after the EOF tag. Whilst this method is simple, a range of steganography software distributed online uses it (Camouflage, JpegX, Data Stash [25]). Unfortunately, this simple technique would not resist any kind of editing to the stego-image nor any attacks by steganalysis experts.

Another naïve implementation of steganography is to append hidden data into the image's extended file information (EXIF), which is a standard used by digital camera manufacturers to store information in the image file, such as, the make and model of a camera, the time the picture was taken and digitized, the resolution of the image, exposure time, and the focal length. This is metadata information about the image and its source located at the header of the file. Special agent Paul Alvarez [26] discussed the possibility of using such headers in digital evidence analysis to combat child pornography. Fig. 9 depicts some text inserted into the comment field of a GIF image header. This method is not a reliable one as it suffers from the same drawbacks as that of the EOF

Fig. 8. The secret message revealed when the stego-image is opened using Notepad. Note that the format of the inserted message remains intact.

Fig. 9. Text insertion into EXIF header: (top) the inserted text string highlighted in a box and (bottom) its corresponding hexadecimal chunk.

method. Note that it is not always recommended to hide data directly without encrypting as in this example.

3.2. Steganography in the image spatial domain

In spatial domain methods a steganographer modifies the secret data and the cover medium in the spatial domain, which involves encoding at the level of the LSBs. This method although simpler, has a larger impact compared to the other two types of methods [26].

A general framework showing the underlying concept is highlighted in Fig. 10. A practical example of embedding in the 1st LSB and up to the 4th LSB is illustrated in Fig. 11. It can be seen that embedding in the 4th LSB generates more visual distortion to the cover image as the hidden information is seen as "non-natural".

It is apparent to an observer that Fig. 11 concludes that there is a trade-off between the payload and the cover image distortion; however the payload, (embedding up to the 1st, 2nd, 3rd, or 4th LSB) is analogous with respect to the recovered embedded image. For instance, Fig. 11(k) (recovered from embedding into 4 LSBs) is a good estimate of the hidden image (Fig. 11(c)) but produces noticeable artefacts (Fig. 11(f)). On the other hand (Fig. 11(j)) (recovered from embedding into 1st LSB) trades bad quality with an almost identical carrier to the original (compare Fig. 11(d) with Fig. 11(a)).

Potdar et al. [27] used a spatial domain technique in producing a fingerprinted secret sharing steganography for robustness against image cropping attacks. Their paper addressed the issue of image cropping effects rather than proposing an embedding technique. The logic behind their proposed work is to divide the cover image into sub-images and compress and encrypt the secret data. The resulting data is then sub-divided in turn and embedded into those image portions. To recover the data, a Lagrange Interpolating Polynomial was applied along with an encryption algorithm. The computational load was high, but their algorithm parameters, namely the number of sub-images (n) and the threshold value (k) were not set to optimal values leaving the reader to guess the values. Bear in mind also that if n is set to 32, for example, that means 32 public keys are needed along with 32 persons and 32 sub-images, which turns out to be unpractical. Moreover, data redundancy that they intended to eliminate does occur in their stego-image.

Shirali-Shahreza and Shirali-Shahreza [28] exploited Arabic and Persian alphabet punctuations to hide messages. While their method is not related to the LSB approach, it falls into the spatial domain if the text is treated as an image. Unlike the English which has only two letters with dots in their lower case format, namely "*i*" and "*j*", Persian language is rich in that 18 out of 32 alphabet letters have dots. The secret message is binarized and those 18 letters' dots are modified according to the values in the binary file.

Colour palette based steganography exploits the smooth ramp transition in colours as indicated in the colour palette. The LSBs here are modified based on their positions in the palette index. Johnson and Jajodia [1] were in favour of using BMP (24 bit) instead of JPEG images. Their next-best choice was GIF files (256-color). BMP as well as GIF based steganography apply LSB techniques, while their resistance to statistical counter-attacks and compression are reported to be weak [3,29–32]. BMP files are bigger compared to other formats which render them improper for network transmissions. JPEG images however, were at the beginning avoided because of their compression algorithm which does not support a direct LSB embedding into the spatial domain. In [33], the authors claimed that changes as small as flipping the LSB of one pixel in a JPEG image can be reliably detected. The experiments on the discrete cosine transform (DCT) coefficients showed promising results and redirected researchers' attention towards this type of image. In fact acting at the level of DCT makes steganography more robust and less prone to statistical attacks.

Jung and Yoo [34] down-sampled an input image to $\frac{1}{2}$ of its size and then used a modified interpolation method, termed the neighbour mean interpolation (NMI), to up-sample the result back to its original dimensions ready for

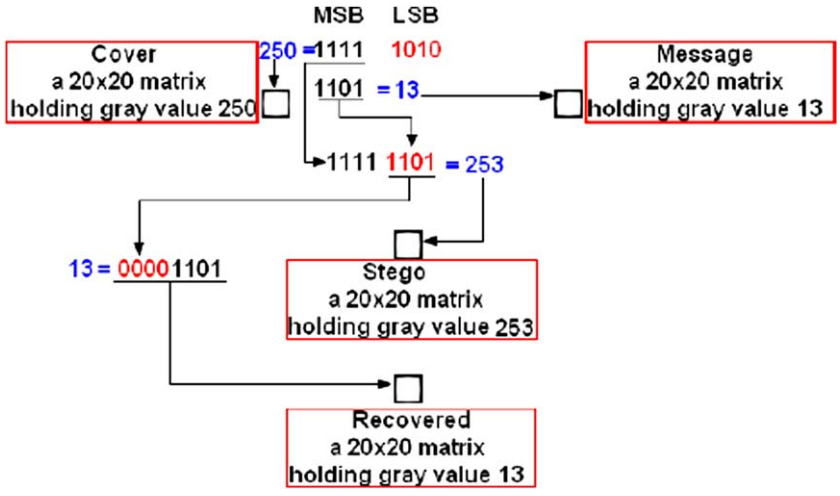

Fig. 10. Steganography in spatial domain. The effect of altering the LSBs up to the 4th bit plane.

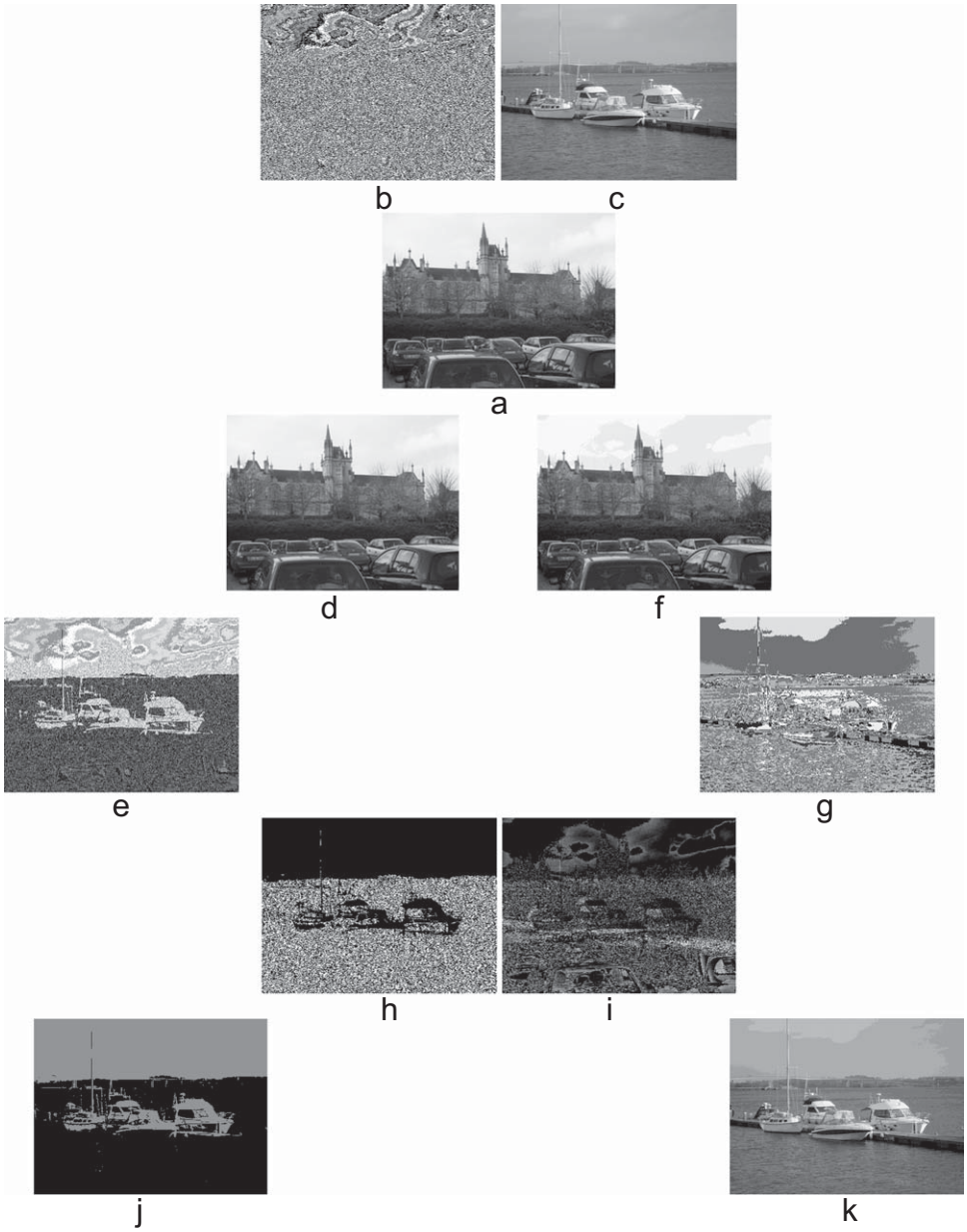

Fig. 11. A plain (without encryption or pre-processing) implementation of steganography in the spatial domain. (a) The cover carrier—University of Ulster, (b) 1st–4th LSBs of (a) with the contrast being enhanced for better visualization, (c) the image to hide—Londonderry's river-, (d) stego-image 1st LSBs replaced with 1st MSBs of (c), (e) LSBs of (d), (f) stego-image 1st–4th LSBs replaced with 1st–4th MSBs of (c), (g) LSBs of (f), (h) difference between (a) and (d), (i) difference between (a) and (f), (j) hidden image extracted from (d), (k) hidden image extracted from (f).

embedding. For the embedding process the up-sampled image was divided into 2×2 non-overlapping blocks as shown in Fig. 12. Potential problems with this method are:

- the impossibility of recovering the secret bits without errors, owing to the use of $\log_2$, which is also used in the extraction that produces floating point values, and
- since in the 222 blocks, the leading value (i.e., block(1,1)) is left unaltered, thus this would lead to the destruction of the natural strong correlation between adjacent pixels which would advertise a non-natural process involvement

Histogram-based data hiding is another commonly used data hiding scheme. Li et al. [35] propose lossless data hiding using the difference value of adjacent pixels. It is classified under "± 1" data embedding algorithms. It exploits the correlation between adjacent pixels that eventually results in a compact histogram that is characterized by a normal Gaussian distribution (as

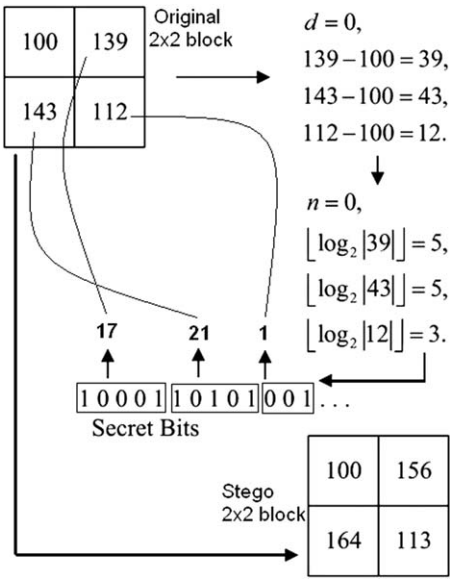

Fig. 12. The system reported in Jung and Yoo [34].

shown in Fig. 13). Instead of considering the whole image, Piyu Tsai et al. [36] divide the image into blocks of 5×5 where the residual image is calculated using linear prediction (another term for adjacent pixels' difference). Then the secret data is embedded into the residual values, followed by block reconstruction.

Such schemes have the advantage of recovering the original cover image from the stego-image. While this preservation can be required in certain applications such as medical imaging, in general steganography is not concerned with such recovery. The hiding capacity is restricted in these methods, besides the "± 1" embedding strategy can be detected (see for example Cancelli et al. [37]).

3.3. Steganography in the image frequency domain

New algorithms keep emerging prompted by the performance of their ancestors (spatial domain methods), by the rapid development of information technology and by the need for an enhanced security system. The discovery of the LSB embedding mechanism is actually a big achievement. Although it is perfect in not deceiving the HVS, its weak resistance to attacks left researchers wondering where to apply it next until they successfully applied it within the frequency domain.

The description of the two-dimensional DCT for an input image F and an output image T is calculated as:

$$T_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N}, \quad (3)$$

where

$0 \le p \le M-1$

$0 \le q \le N-1$

and

$$\alpha_p = \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 1 \le p \le M-1 \end{cases} \alpha_q = \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 1 \le q \le N-1 \end{cases}$$

where M, N are the dimensions of the input image while m, n are variables ranging from 0 to $M-1$ and 0 to $N-1$ respectively.

DCT is used extensively with video and image compression e.g. JPEG lossy compression. Each block DCT coefficients obtained from Eq. (3) are quantized using a specific quantization table (QT). This matrix shown in Fig. 14 is suggested in the Annex of the JPEG standard, note that some camera manufacturers have their own built-in QT and they do not necessarily conform to the standard JPEG table. The logic behind choosing a table with such values is based on extensive experimentation that tried to balance the trade-off between image compression and quality factors. The HVS dictates the ratios between values in the QT.

The aim of quantization is to loosen up the tightened precision produced by DCT while retaining the valuable information descriptors. The quantization step is specified by:

$$f'(\omega_x, \omega_y) = \left\lfloor \frac{f(\omega_x, \omega_y)}{\Gamma(\omega_x, \omega_y)} + \frac{1}{2} \right\rfloor, \quad \omega_x, \omega_y \in 0, 1, \ldots, 7 \quad (4)$$

where x and y are the image coordinates, $f'(\omega_x, \omega_y)$ denotes the result function, $f(\omega_x, \omega_y)$ is an 8×8 non-overlapping intensity image block and $\lfloor . \rfloor$ a floor rounding operator. $\Gamma(\omega_x, \omega_y)$ represents a quantization step which, in relationship to JPEG quality, is given by:

$$\Gamma(\omega_x, \omega_y) = \begin{cases} \max\left(\left\lfloor \frac{200 - 2Q}{100} QT(\omega_x, \omega_y) + \frac{1}{2} \right\rfloor, 1 \right), & 50 \le Q \le 100 \\ \left\lfloor \frac{50}{Q} QT(\omega_x, \omega_y) + \frac{1}{2} \right\rfloor, & 0 \le Q \le 50 \end{cases} \quad (5)$$

where $QT(\omega_x, \omega_y)$ is the quantization table depicted in Fig. 14 and Q is a quality factor. JPEG compression then applies entropy coding such as the Huffman algorithm to compress the resulted $\Gamma(\omega_x, \omega_y)$. Most of the redundant data and noise are lost in this stage hence the name lossy compression. For more details on JPEG compression the reader is directed to Popescu's work [38].

The above scenario is a discrete theory independent of steganography. Li and Wang [39] presented a steganographic method that modifies the QT and inserts the hidden bits in the middle frequency coefficients. Their modified QT is shown in Fig. 15. The new version of the QT gives them 36 coefficients in each 8×8 block to embed their secret data into which yields a reasonable payload. Their work was motivated by a prior published work [40]. Steganography based on DCT JPEG compression goes through different steps as shown in Fig. 16.

Most of the techniques here use JPEG images as vehicles to embed their data. JPEG compression uses the DCT to transform successive sub-image blocks (8×8 pixels) into 64 DCT coefficients. Data is inserted into these coefficients' insignificant bits; however, altering any single coefficient would affect the entire 64 block pixels [41]. As the change is operating on the frequency domain

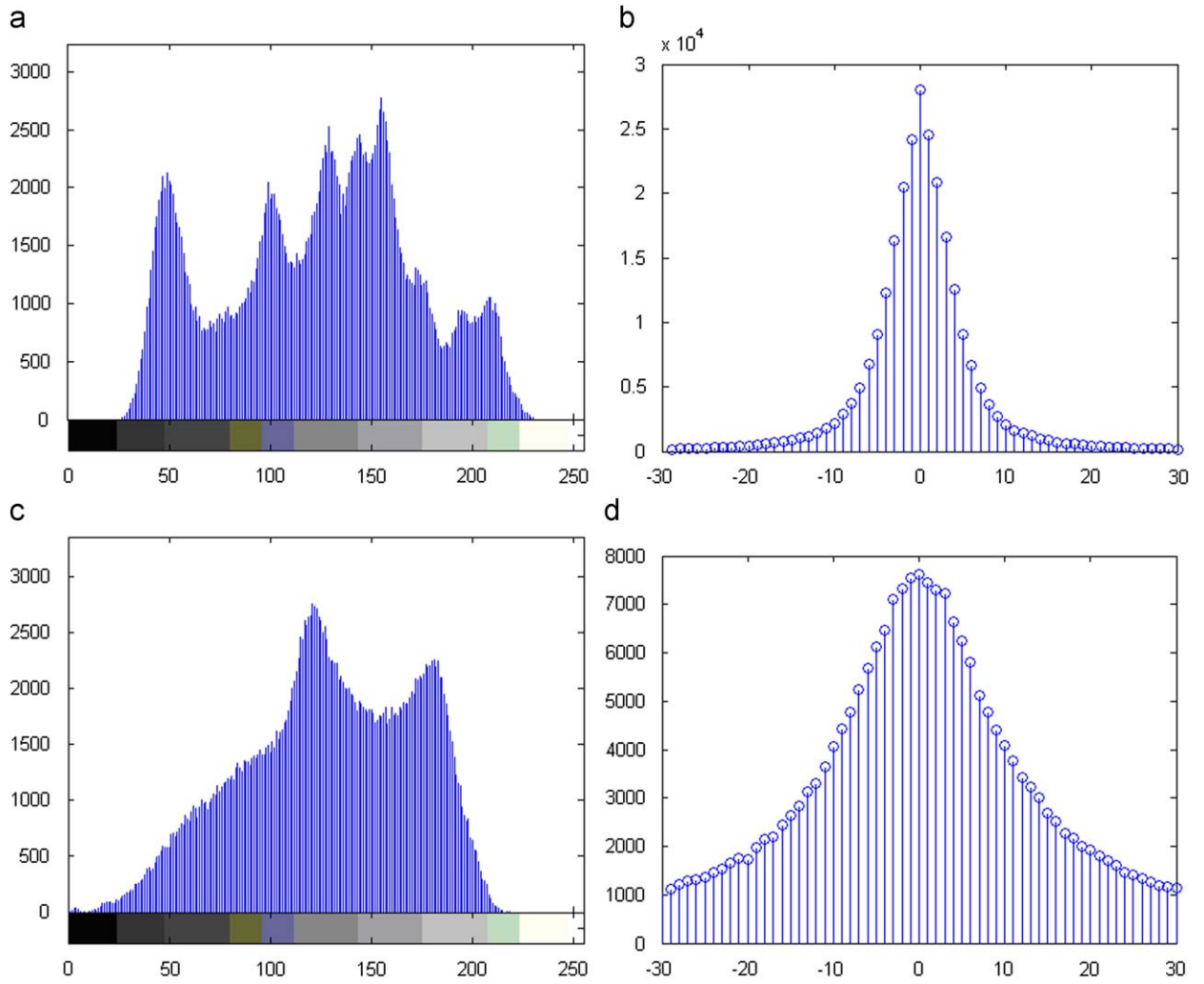

Fig. 13. Histograms of Lena and Baboon: (a) histogram of Lena; (b) difference histogram of Lena; (c) histogram of Baboon; (d) difference histogram of Baboon [36].

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Fig. 14. JPEG suggested Luminance Quantization Table used in DCT lossy compression. The value 16 (in bold-face) represents the DC coefficient and the other values are the AC coefficients.

8	1	1	1	1	1	1	1
1	1	1	1	1	1	1	55
1	1	1	1	1	1	69	56
1	1	1	1	1	87	80	62
1	1	1	1	68	109	103	77
1	1	1	64	81	104	113	92
1	1	78	87	103	121	120	101
1	92	95	98	112	100	103	99

Fig. 15. The modified Quantization Table [39].

instead of the spatial domain there will be no visible change in the cover image given those coefficients are handled with care [42].

According to Raja et al. [43] fast Fourier transform (FFT) methods introduce round-off errors; thus it is not suitable for hidden communication. However, Johnson and Jajodia [1], thought differently and included it among the used transformations in steganography and McKeon [44] utilised the 2D discrete Fourier transform (DFT) to generate Fourier based steganography in movies.

Choosing which values in the 8×8 DCT coefficients block are altered is very important as changing one value will affect the whole 8×8 block in the image. Fig. 17 shows a poor implementation of such a method in which careful consideration was not given to the sensitivity of DCT coefficients.

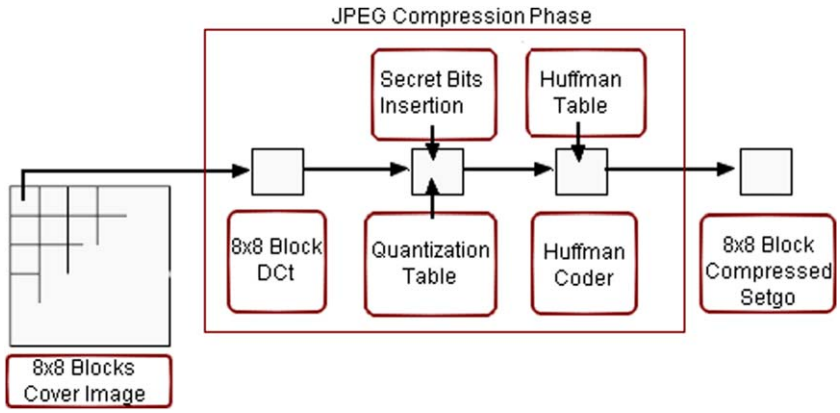

Fig. 16. Data flow diagram showing the general process of embedding in the frequency domain.

Fig. 17. Embedding at the DCT level is a very successful and powerful tool—but if coefficients are not carefully selected some artefacts will be noticeable.

The JSteg algorithm was among the first algorithms to use JPEG images. Although the algorithm stood strongly against visual attacks, it was found that examining the statistical distribution of the DCT coefficients shows the existence of hidden data [3]. JSteg is easily detected using the X^2-test. Moreover, since the DCT coefficients need to be treated with sensitive care and intelligence the JSteg algorithm leaves a significant statistical signature. Wayner [45] stated that the coefficients in JPEG compression normally fall along a bell curve and the hidden information embedded by JSteg distorts this. Manikopoulos et al. [46] discussed an algorithm that utilises the probability density function (PDF) to generate discriminator features fed into a neural network system which detects hidden data in this domain.

OutGuess [3] was a better alternative as it used a pseudo-random-number generator to select DCT coefficients. The X^2-test does not detect data that is randomly distributed. The developer of OutGuess suggests a counter-attack against his algorithm. Provos et al. [3,47,48] suggest applying an extended version of the X^2-test to select Pseudo-randomly embedded messages in JPEG images.

Andreas Westfeld based his "F5" algorithm [49] on subtraction and matrix encoding (also known as syndrome coding). F5 embeds only into non-zero AC DCT coefficients by decreasing the absolute value of the coefficient by 1. A shrinkage occurs, as described in [50], when the same bit has to be re-embedded in case the original coefficient is either "1" or "−1" as at the decoding phase all zero coefficients will be skipped whether they were modified or not. Neither X^2-test nor its extended versions could break this solid algorithm. Unfortunately, F5 did not survive attacks for too long. Fridrich et al. [33] proposed steganalysis that does detect F5 contents, disrupting F5's survival.

Another trend related to the above quantization table modification (Fig. 15) is the so-called perturbed quantization (PQ) [51], which aims to achieve high efficiency, with minimal distortion, rather than a large capacity. Each coefficient in the DCT block is assigned a scalar value that corresponds to how much impact it would make to the carrier image, and then a steganographer can set a selection rule to filter out the "well behaved" coefficients, thus giving the algorithm less payload but high imperceptibility.

As for steganography in the discrete wavelet transform (DWT), the reader is directed to some examples in the literature [52–54]. Abdulaziz and Pang [55] use vector quantization called Linde-Buzo-Gray (LBG) coupled with block codes known as BCH code and 1-stage discrete Haar wavelet transforms. They reaffirm that modifying data using a wavelet transformation preserves good quality with little perceptual artefacts.

The DWT-based embedding technique is still in its infancy. Paulson [56] reports that a group of scientists at Iowa State University are focusing on the development of an innovative application which they call "Artificial Neural Network Technology for steganography (ANNTS)" aimed at detecting all present steganography techniques including DCT, DWT and DFT. The inverse discrete Fourier transform (iDFT) encompasses round-off error which renders DFT improper for steganography applications.

Abdelwahab and Hassan [57] propose a data hiding technique in the DWT domain. Both secret and cover images are decomposed using DWT (1st level). Each of

which is divided into disjoint 4×4 blocks. Blocks of the secret image fit into the cover blocks to determine the best match. Afterwards, error blocks are generated and embedded into coefficients of the best matched blocks in the HL of the cover image. Two keys must be communicated; one holds the indices to the matched blocks in the CLL (cover approximation) and another for the matched blocks in the CHL of the cover. Note that the extracted payload is not totally identical to the embedded version as the only embedded and extracted bits belong to the secret image approximation while setting all the data in other sub-images to zeros during the reconstruction process.

3.4. Adaptive steganography

Adaptive steganography is a special case of the two former methods. It is also known as "Statistics-aware embedding" [3], "Masking" [1] or "Model-Based" [58]. This method takes statistical global features of the image before attempting to interact with its LSB/DCT coefficients. The statistics will dictate where to make the changes [59,60]. It is characterized by a random adaptive selection of pixels depending on the cover image and the selection of pixels in a block with large local STD (standard deviation). The latter is meant to avoid areas of uniform colour (smooth areas). This behaviour makes adaptive steganography seek images with existing or deliberately added noise and images that demonstrate colour complexity. Wayner [45] dedicated a complete chapter in a book to what he called "life in noise", pointing to the usefulness of data embedding in noise. It is proven to be robust with respect to compression, cropping and image processing [41,61,62]. The model-based method (MB1), described in [58], generates a stego-image based on a given distribution model, using a generalized Cauchy distribution, that results in the minimum distortion. Due to the lack of a perfect model, this steganographic algorithm can be broken using the first-order statistics [63]. Moreover, it can also be detected by the difference of "blockiness" between a stego-image and its estimated image reliably [64]. The discovery of "blockiness" led the author in [58] to produce an enhanced version called MB2, a model-based with de-blocking. Unfortunately, even MB2 can be attacked as highlighted in Section 5.

Edge embedding follows edge segment locations of objects in the host gray scale image in a fixed block fashion each of which has its centre on an edge pixel. Whilst simple, this method is robust to many attacks and it follows that this adaptive method is also an excellent means of hiding data while maintaining a good perceptibility.

Chin-Chen et al. [65], propose an adaptive technique applied to the LSB substitution method. Their idea is to exploit the correlation between neighbouring pixels to estimate the degree of smoothness. They discuss the choices of having 2–4 sided matches. The payload (embedding capacity) was high.

Hioki [66], presented an adaptive method termed "A Block Complexity based Data Embedding" (ABCDE). Embedding is performed by replacing selected suitable pixel data of noisy blocks in an image with another noisy block obtained by converting data to be embedded. This suitability is identified by two complexity measures to properly discriminate complex blocks from simple ones; which are run-length irregularity and border noisiness (see Fig. 18). The hidden message is more a part of the image than being added noise [67]. The ABCDE method introduced a large embedding capacity; however, certain control parameters had to be configured manually, e.g., finding an appropriate section length for sectioning a stream of resource blocks and finding the threshold value that controls identification of complex blocks. These requirements render the method unsuitable for automatic processes. Table 2 shows the parameters that the algorithm encompasses. To get rid of fake complex

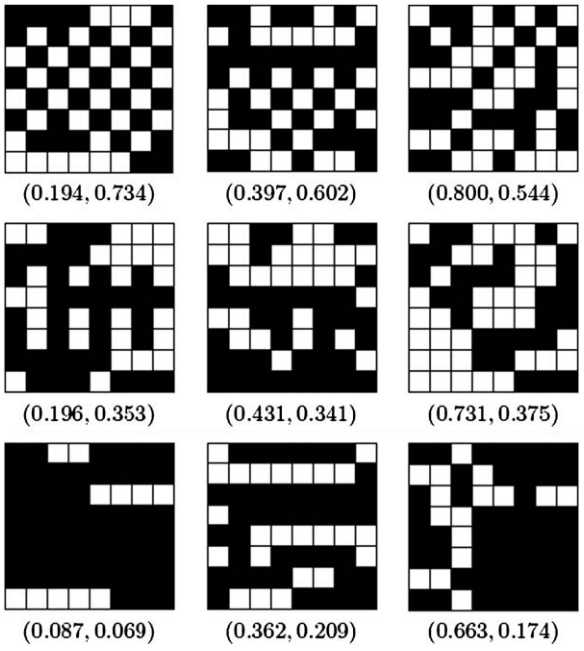

Fig. 18. Blocks of various complexity values (β for run-length irregularity, γ for border noisiness) [66].

Table 2
Parameters of ABCDE [66].

External Parameters
Block size ($n \times n$)
External or Internal Parameters
M-sequence parameters
The characteristic polynomial
The initial polynomial
The seed
Threshold values for complexity measures for each bit plane
Internal Parameters
Resource file parameters
The name of the resource file
The size of the resource file
The length of sections

blocks resulting from considering an adjacent pure binary code (PBC), Hioki chooses to convert decimals into reflected binary gray code (RBGC). The problem which RBGC was used to solve was the complexity of the higher bit planes to tolerate little relation to the true variation of the image pixels' intensities creating what is often called "hamming cliffs" [68].

There are two vague issues which are obscurely discussed at the end of Hioki's work. One arises when the carrier image's dimensions are not proportional to the block division scheme and so fragments from these dimensions are kept away from the embedding process. There was no indication by the author of the possible impact of this decision as it might leave a clear contrast between the modified and the intact parts of the image which distorts its statistical properties. The second point is the introduction of the zero padding when the compressed resource file size is not a multiple of the block size. The author did not show any explanation on how to generate complexity from such a compressed file since there will be a sequence of zeros resulting from the "0" padding notion. The author in the experimental section does not show how resilient the algorithm is to different image processing attacks, e.g., rotation, additive noise, cropping, and compression. Indeed, the ABCDE algorithm provides an improvement over a former method known as BPCS (bit plane complexity segmentation) [69]; which, in turn, was introduced to compensate for the drawback of the traditional LSB manipulation techniques of data hiding [70]. The computational complexity of the algorithm to find a phase key that passes the threshold is time consuming and there is no guarantee that it will always evolve into an optimal solution [71]. BPCS steganography is not robust to even small changes in the image [72], and this weakness is inherited by the ABCDE algorithm also since its underlying framework is based on BPCS. This intolerance to any manipulation of the stego-image is perceived by the authors in [72] as a merit. They were over-optimistic about this lack of robustness in the sense that any kind of attack would "destroy the embedded evidence" which points, in their view, to image tampering. Robustness of steganography is one of the three main goals to be achieved and this is definitely not shown in Kawaguchi's argument. Their algorithm would fail to retrieve the embedded data in two cases: first when the stego-image is attacked resulting in the destruction of the embedded data, and second when an image is plain clear (meaning that no embedding process took place). These two contradictory justifications, due primarily to lack of robustness, would not be appealing characteristics to forensics experts or other interested bodies.

In [67], the authors chose to use wavelet transforms that map integers to integers instead of using the conventional wavelet Transforms. This can overcome the difficulty of floating point conversion that occurs after embedding. Their scheme embeds the payload in non overlapping 4×4 blocks of the low frequency, where two pixels at a time are chosen, one on either side of the principal diagonal. Cover image adjustment was required to prevent the problem of under/overflow of pixel values after embedding. In the respective section, they discuss the overflow problem only, where they suggest using the following system prior to embedding:

$$C'(i,j,k) = \begin{cases} C(i,j,k) - (2^N - 1) & \text{if } C(i,j,k) = 255 \\ C(i,j,k) & \text{Otherwise.} \end{cases} \quad (6)$$

where $C'(i,\ j,\ k)$ denotes the modified pixel and N represents the number of bits to be embedded in each coefficient (i.e., N=4). This means any value of 255 will be converted to 240. For a true colour image format, they apply the algorithm on each colour plane separately. This step ignores the high correlation between colour planes in natural images. Not taking this phenomenon into consideration means the embedding scenario will corrupt some of the inherited statistics of the cover image, a trap that severely exposes the stego-image to steganalysis attacks. The authors also state some assumptions; first, embedding is carried out only on non-singular matrices, also ± 15 is imperceptible to human vision; finally, the cover image and payload are assumed to be JPEG and the cover be a square matrix of size 512×512. We doubt the second assertion however. Even though this can be possibly acceptable from a human visual perspective, however, from a statistical point of view, this amount of change is intolerable. Before they conclude, they state that their cover image and stego-image version are similar, even though the best candidate in their experiments has a PSNR that did not exceed 45.

In [73], the authors attempt to create a method to restore the marked image to its pristine state after extracting the embedded data. They achieve this by applying the pick point of a histogram in the difference image to generate an inverse transformation in the spatial domain. The cover image is divided into non-overlapping 4×4 blocks where a difference matrix of size 3×4 is generated for each block. The selection of the local histogram's peak point p_b will direct the embedding process and matrix manipulation. The example shown in their hiding phase section might not be sufficient to verify the accuracy of the algorithm. Some questions remain unanswered such as what happens when we have two peak points instead of one? On which criterion will we base our selection? Another issue occurs when transforming the matrix SD_b to RD_b; it is highly likely that after the subtraction process we will have some values that collude with the peak value which confuses the extraction of the embedded data. To prevent over/underflow, caused by the arithmetic operations on values close to boundaries (i.e., 0, 255), the authors use the modulus operator (i.e., mod 256). There was no adequate explanation on the effect of homogeneous, dark, bright, and edged blocks on the algorithm efficiency.

In [74], a GA-based algorithm is presented which generates a stego-image to break the detection of the spatial domain and the frequency-domain steganalysis systems by artificially counterfeiting statistical features. Time complexity, which is usually the drawback of genetic based algorithms, was not discussed though. They mentioned that "the process is repeated until a predefined condition is satisfied or a constant number of iterations

are reached. The predefined condition is the situation when we can correctly extract the desired hidden message." Again, it was not stated whether the process of determining such a condition was done automatically or involved a human inference (visual perception). The suggested GA-based rounding-error correction algorithm, whilst interesting, still needs proof of generalization. Wu and Shih [74] closed their introduction section by saying, "this is the first paper of utilizing the evolutionary algorithms in the field of steganographic systems". It should be noted that image hiding using genetic algorithm was known prior to their work such as the work in [75]. In [64], the authors proposed extending the conventional "± 1" algorithm to JPEG images using genetic algorithm.

Kong et al. [76] proposed a content-based image embedding based on segmenting homogenous grayscale areas using a watershed method coupled with Fuzzy C-Means (FCM). Entropy was then calculated for each region. Entropy values dictated the embedding strength where four LSBs of each of the cover's RGB primaries were used if it exceeded a specific threshold otherwise only two LSBs for each were used. The drawback of this method was its sensitivity to intensity changes which would affect severely the extraction of the correct secret bits. As a side note, Kong et al. [76] also reported the use of a logistic map to encrypt the secret bit stream which seems venerable to a Chosen-plaintext attack (CPA).

Chao et al. [77] presented a 3D steganography scheme. The embedding scheme hides secret messages in the vertices of 3D polygon models. Similarly, Bogomjakov et al. [78], hide a message in the indexed representation of a mesh by permuting the order in which faces and vertices are stored. Although, such methods claim higher embedding capacity, however time complexity to generate the mesh and then rendering can be an issue. Moreover 3D graphics are not that portable compared to digital images.

Nakamura and Zhao [79], propose a morphing process that takes as input the secret image and the cover file. The method does not discuss the generated features from the cover and secret images used for morphing and how to regenerate them from the stego-image.

Zeki and Azizah [80] proposed what they termed as "the intermediate significant bit algorithm". They studied different ranges of an 8 bit image and found the best compromise for distortion and robustness was in the following range: [0:15] [16:31] … [224:239] [240:255]. The core idea in the embedding process is to find the nearest range that matches the secret bit in the next or previous range.

4. Analysis and recommendations

As a performance measurement for image distortion, the well known peak-signal-to-noise ratio (PSNR) which is classified under the difference distortion metrics can be applied on the stego-images. It is defined as:

$$PSNR = 10\log10\left(\frac{C_{\max}^2}{MSE}\right) \tag{7}$$

where *MSE* denotes mean square error which is given as:

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(S_{xy} - C_{xy})^2 \tag{8}$$

where x and y are the image coordinates, M and N are the dimensions of the image, S_{xy} is the generated stego-image and C_{xy} is the cover image. Also $C_{\max}^2$ holds the maximum value in the image, for example:

$$C_{\max}^2 \leq \begin{cases} 1, & \text{double-precision} \\ 255, & \text{uint8 bit} \end{cases}$$

Many authors [39,42,81–84], consider $C_{\max}$=255 as a default value for 8 bit images. It can be the case, for instance, that the examined image has only up to 253 or fewer representations of gray colours. Knowing that $C_{\max}$ is raised to a power of 2 results in a severe change to the PSNR value. Thus $C_{\max}$ can be defined as the actual maximum value rather than the largest possible value. PSNR is often expressed on a logarithmic scale in decibels (dB). PSNR values falling below 30 dB indicate a fairly low quality, i.e., distortion caused by embedding can be obvious; however, a high quality stego-image should strive for 40 dB and above.

Van Der Weken et al. [85] proposed other similarity measures (*SMs*). They analysed the efficiency of ten SMs in addition to a modified version of PSNR constructed based on neighbourhood blocks which better adapt to human perception. In order to produce a fair performance comparison between different methods of invisible watermarking, Kutter and Petitcolas [86] discussed a novel measure adapted to the human visual system.

Table 3 shows different PSNR values spawned by various software based on spatial domain method described in Section 3.2 [25], applied on the images shown in Figs. 19–22 (which depict the output of each of the tools).

It is also noted that some algorithms, like the one used in the Revelation software, have the pair effect fingerprint that appears on stego-images.

Table 4 compares some software tools appearing in [25]. We based our comparison on the following factors:

- the domain on which the algorithm is applied, e.g., spatial or frequency domain,
- the support for encryption,
- random bit selection and
- the different supported image formats.

A performance analysis of some steganographic tools is provided in [59]. The drawback of the current techniques is tabulated in Table 5.

There appears to be two main groups in the area, one for creating steganography algorithms and another group for creating a counter-attack (steganalysis). Fard et al. [41] state clearly that "there is currently no steganography system which can resist all steganalysis attacks". "Ultimately, image understanding is important for secure adaptive steganography. A human can easily recognize that a pixel is actually a dot above the letter "*i*" and must not be changed. However, it would be very hard to write a

Table 3
Summary of performance of common software [59].

Software	PSNR		Visual inspection
	Set A	Set B	
[Hide&Seek]	18.608	22.7408	Very clear grainy noise in the stego-image, which renders it the worst performer in this study
[Hide-in-Picture]	23.866	28.316	Little noise. Accepts only 24 bit bmp files. Creates additional colour palette entries. In this case the original boat image has 32 colours and the generated stego-image augmented the number to 256 by creating new colours
[Stella]	26.769	16.621	Little noise. Works only with 24 bit images
[S-Tools]	37.775	25.208	No visual evidence of tamper
[Revelation]	23.892	24.381	No visual evidence of tamper, but pair effect appears on the histogram of some outputs

Fig. 19. Images used to generate Tables 2. (Left to right) Set A: cover image Boat, (321×481) and the secret image Tank, (155×151). Set B: cover image Lena 320×480, secret image Male (77×92).

computer program capable of making such intelligent decisions in all possible cases, [70]". "While there are numerous techniques for embedding large quantities of data in images, there is no known technique for embedding this data in a manner that is robust in light of the variety of manipulations that may occur during image manipulation" [15].

"Some researchers proposed to model the cover characteristics and thus create an adaptive steganography algorithm, a goal which is not easily achieved" [87]. Determining the maximal safe bit-rate that can be embedded in a given image without introducing statistical artifacts remains a very complicated task [88]. The above challenges motivated the steganography community to create a more fundamental approach based on universal properties and adaptive measures [89].

In the table, the sign (✓) indicates the characteristic is present, (−) denotes unavailability of information at present, while (×) gives the negative response. As it is clear from the table, all of the mentioned steganographic algorithms have been detected by steganalysis methods and thus a robust algorithm with a high embedding capacity needs to be investigated.

Based on the literature the following points are noted:

- Algorithms F5 and Outguess are the most reliable although they violate the second order statistics. Both utilise DCT embedding.
- Embedding in the DWT domain shows promising results and outperforms DCT embedding especially in terms of compression survival [45]. A steganographer should be cautious when embedding in the transformation domains in general; however DWT tends to be more flexible than DCT. Unlike JPEG, the introduced image coding system JPEG2000[4] allows wavelets to be employed for compression in lieu of the DCT. This makes DWT based steganography the future leading method.
- Without loss of generality; edge embedding maintains an excellent distortion free output whether it is applied in the spatial, DCT or DWT domains [90]. However, the limited payload is its downfall.

Recognising and tracking elements in a given carrier while embedding can help survive major image processing attacks and compression. This manifests itself as an adaptive intelligent type where the embedding process affects only certain regions of interest (ROI) rather than the entire image. With the boost of computer vision (CV) and pattern recognition disciplines this method can be fully automated and unsupervised. These elements (ROIs), e.g., faces in a crowd [91], can be adjusted in perfectly undetectable ways. The majority of steganography research to date has overlooked the fact that utilising objects within images can strengthen the embedding robustness—with few exceptions. A steganography approach reported in [92,93], incorporated computer vision to track and segment skin regions for embedding under the assumption that skin tone colour provides better embedding imperceptibility. They used computer vision techniques to introduce their rotation and translation invariance embedding scheme to establish an object oriented embedding (OOE). A related method, in the sense that it uses objects in images although it is meant for watermarking instead, was introduced by authors in [94,95] where they employed an adaptive clustering technique in order to derive a robust region representation of the original image. The robust regions were approximated by ellipsoids, whose bounding rectangles were chosen as the embedding area for the watermark.

Most of the existing steganographic methods rely on two factors: the secret key and the robustness of the steganographic algorithm. However, all of them either do not address the issue of encryption of the payload prior to embedding or merely give a hint of using one or more of the conventional block cipher algorithms. Hence, Westfeld et al. concluded their CRYSTAL project with an important observation that "Crypto–Stego interaction is not very well researched, yet".[5] Authors of [96,97] are among those

[4] JPEG2000, available from: http://www.jpeg.org/jpeg2000/.

[5] The CRYSTAL project, [Online]. Available from: ⟨http://www1.inf.tu-dresden.de/~aw4/crystal/slides.slide_1.html⟩.

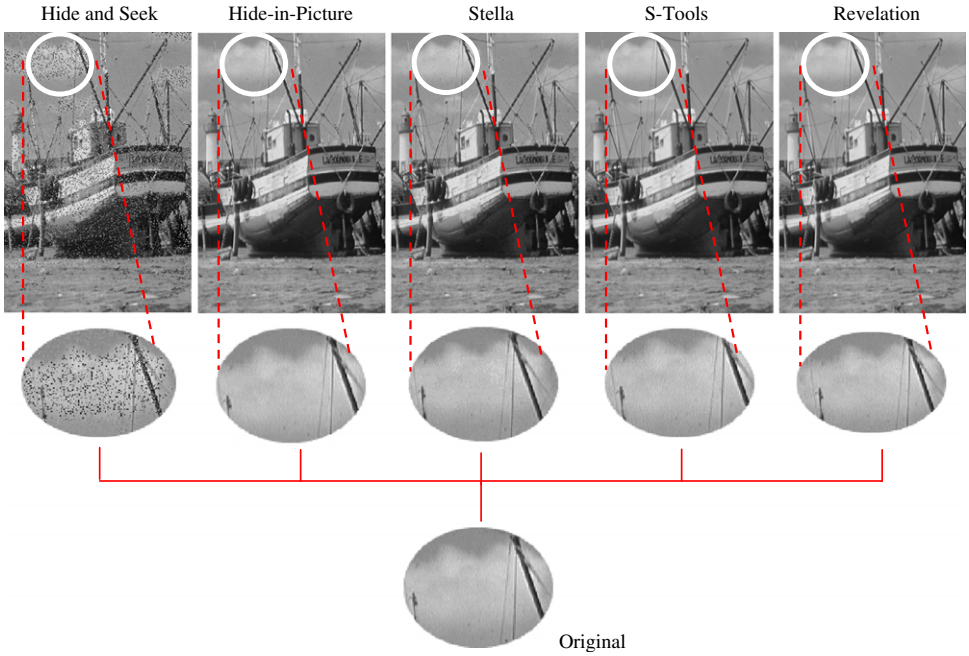

Fig. 20. Set A: stego-images of each tool appearing in Table 3.

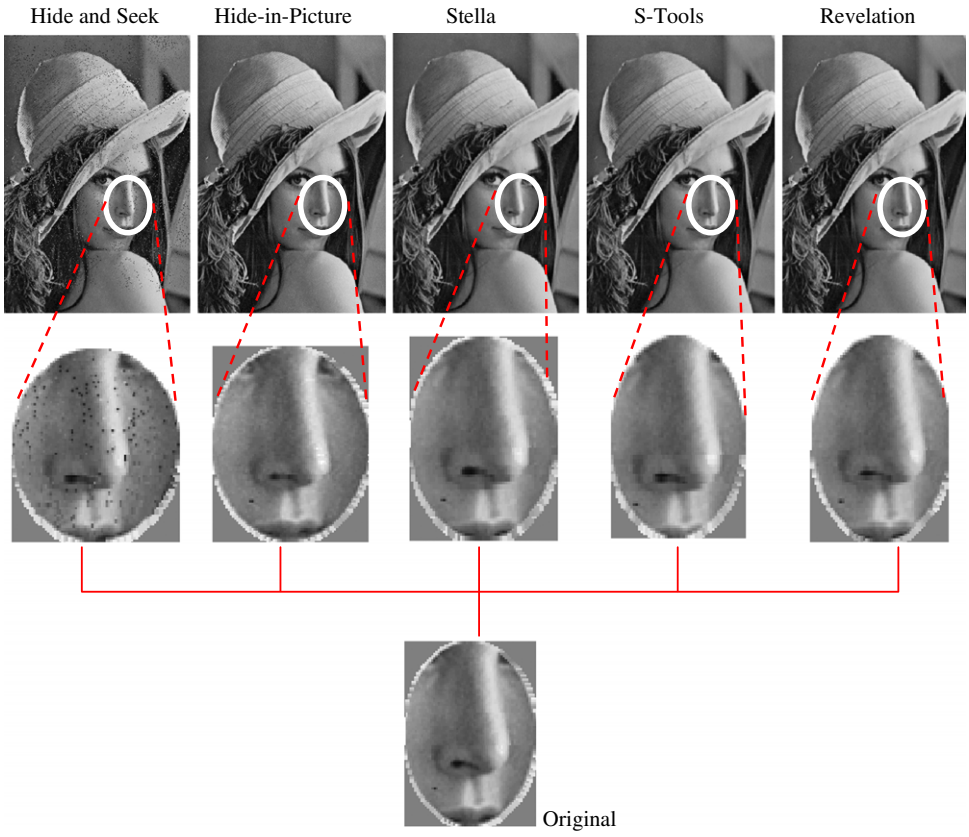

Fig. 21. Set B: stego-images of each tool appearing in Table 3.

who discuss in details the encryption of the payload prior to embedding.

There are some basic notes that should be observed by a steganographer:

- In order to eliminate the attack of comparing the original image file with the stego-image, where a very simple kind of steganalysis is essential, we can freshly create an image and destroy it after generating the stego-image. Embedding into images available on the World Wide Web is not advisable as a steganalysis devotee might notice and opportunistically utilize them to decode the stego-image.
- In order to avoid any Human Visual Perceptual attack, the generated stego-image must not have visual artifacts. Alteration made up to the 4th LSB of a given pixel will yield a dramatic change in its value. Such unwise choice on the part of the steganographer will thwart the perceptual security of the transmission.

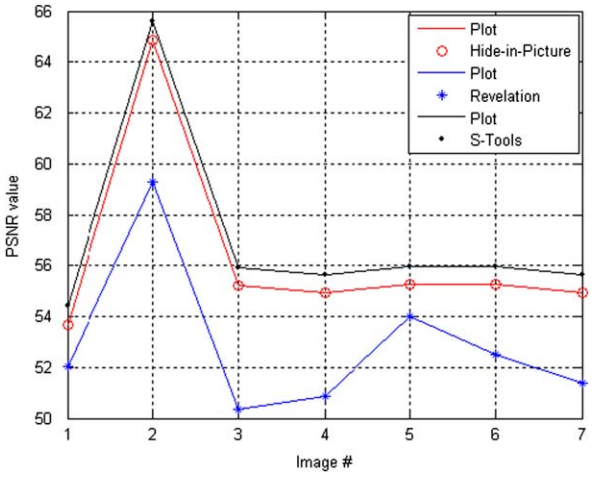

Fig. 22. Additional experiments on steganography software.

Consider the following example: let a pixel intensity value be 173, which in binary is $(10101101)_2$. If the secret bit is "0" then the stego-image pixel will be 165 ($(10100101)_2$ in binary) or 172 ($(10101100)_2$ in binary).

- Smooth homogeneous areas must be avoided, e.g., cloudless blue sky over a blanket of snow; however chaotic areas with naturally redundant noisy backgrounds and salient rigid edges should be targeted [23,98]. This point, however, needs further investigation as some authors think differently. An example is the study of Kodovsky and Fridrich [99] that concludes "texture-adaptive selection channels do not improve steganographic security".
- The secret data must be a composite of a balanced bit values, since in general, the expected probabilities of bit 0 and bit 1 for a typical cover image are the same (i.e., $\text{Probability}\{0\} = \text{Probability}\{1\} = 0.5$) [100]. In some cases, encryption provides such a balance.

It is essential that encryption not only is able to offer such a balance but also is random enough so that it can mimic the LSBs of the cover image. Even though Wayner [45, p. 26] has answered the question "how random is the noise?" qualitatively there are various methods which estimate randomness quantitatively (see [101]). One way to measure such randomness is to use Cross-Covariance as illustrated in Fig. 23.

The last LSB where the stego-value, compared to the plain-value, is unchanged, increased or decreased by one (change by ± 1 in the 1st LSB or ± 4 in the 3rd LSB) eventually leaves traceable statistical violations. Many algorithms to date still use such conventional models either in the spatial domain or the transform domain.

The RBGC allows alteration to even the third LSB (i.e., change by ± 3) in the DWT without much degradation compared to the conventional use of PBC, see Fig. 24 for the graphical structure of both methods. Let a plain-image pixel at the approximation level of a 1st level DWT be the coefficient C and let the secret bit be "0": C=325.09821988712 (Fig. 25).

Table 4
Comparison of different tools: (1) (2) frequency domain (3) encryption support (4) random bit selection (5) image format.

Name	Creator	Year	(1)	(2)	(3)	(4)	(5)	Detected by
JSteg	Derek Upham	–	×	✓ DCT	×	×	JPEG	X^2-test Stegdetect Fridrich's Algorithm
JSteg-Shell	John Korejwa	–	×	✓ DCT	✓ RC4	–	JPEG	X^2-test
OutGuess version 0.13b	Provos and Honeyman	–	×	✓ DCT	✓ RC4	✓	JPEG	X^2-test (extended version) Stegdetect
White Noise Storm	Ray (Arsen) Arachelian	1994	✓	×	✓	✓	PCX	X^2-test
EZStego	Romana Machado	1996	✓	×	✓	×	BMP, GIF	RS-steganalysis
S-Tools	Andrew Brown	1996	✓	×	✓ IDEA, DES, 3DES,MPJ2, NSEA	×	BMP, GIF	X^2-test
JPhide	Allan Latham	1999	×	✓ DCT	✓ Blowfish	×	JPEG	X^2-test Stegdetect
OutGuess version 0.2	Provos and Honeyman	2001	×	✓ DCT	✓ RC4	✓	JPEG	Fridrich's Algorithm
F5	Andreas Westfeld	2001	×	✓	✓	✓	JPEG	Fridrich's Algorithm

Table 5
Drawback of current steganography methods and benefits of the OOE method.

Method	Descriptions
Spatial domain techniques	Large payload but often offset the statistical properties of the image Not robust against lossy compression and image filters Not robust against rotation, cropping and translation Not robust against noise Many work only on the BMP format
DCT based domain techniques	Less prone to attacks than the former methods at the expense of capacity Breach of second order statistics Breach of DCT coefficients distribution Work only on the JPEG format Double compression of the file Not robust against rotation, cropping and translation Not robust against noise Modification of quantization table
Recommended method [93], see also Fig. 25	Object-oriented embedding (OOE) Small embedding space at the benefit of robustness. Resolved by targeting video files Resistance to rotation, translation, cropping and noise impulses No known statistical vulnerabilities Resistance to lossy compression thanks to the DWT Performs better than DCT algorithms in keeping the carrier distortion to the minimum Ability to embed secret data into different orientation, acts as an additional secret key Re-orienting the stego-image to its origin will invoke interpolation, thus providing a mask that fools any statistical attack

RBGC
C_{int}=325, Store=.09821988712
RBGC (C_{int})='111100111'
Steg-image (RBGC (C_{int}))='111100011'
RBGC-to-Decimal='111100011'→322
Steg-image=Concatenate (322, Store)=322.09821988712
Difference ±3 (odd number).

PBC
Bin (C_{int})=$(101000101)_2$
Steg-image (Bin (C_{int}))=$(101000001)_2$
Bin-to-Decimal=$(101000001)_2$→321
Steg-image=Concatenate (321, Store)=321. 09821988712
Difference ±4 (even number).

5. Steganalysis

This article does not delve into the details of the methods of steganalysis although this work presents, herein, a brief description and some standards that a steganographer should usually examine. Steganalysis is the science of attacking steganography in a battle that never ends. It mimics the already established science of Cryptanalysis. Note that steganographers can create a steganalysis system merely to test the strength of their algorithm. Steganalysis is achieved through applying different image processing techniques, e.g., image filtering, rotating, cropping, and translating. More deliberately, it can be achieved by coding a program that examines the stego-image structure and measures its statistical properties, e.g., first order statistics (histograms) or second order statistics (correlations between pixels, distance, direction). JPEG double compression and the distribution of DCT (discrete cosine transform) coefficients can give hints on the use of DCT-based image steganography. Passive steganalysis attempts to destroy any trace of secret communication, without bother to detect the secrete data, by using the above mentioned image processing techniques: changing the image format, flipping all LSBs or by undertaking a severe lossy compression, e.g., JPEG. Active steganalysis however, is any specialized algorithm that detects the existence of stego-images.

Spatial steganography generates unusual patterns such as sorting of colour palettes, relationships between indexed colours and exaggerated "noise", as can seen in Fig. 26, all of which leave traces to be picked up by steganalysis tools. This method is very fragile [102]. "LSB encoding is extremely sensitive to any kind of filtering or manipulation of the stego-image. Scaling, rotation, cropping, addition of noise, or lossy compression to the stego-image is very likely to destroy the message. Furthermore an attacker can easily remove the message by removing (zeroing) the entire LSB plane with very little change in the perceptual quality of the modified stego-image" [29]. Almost any filtering process will alter the values of many of the LSBs [103].

By inspecting the inner structure of the LSBs, Fridrich and her colleagues [105] claimed to be able to extract hidden messages as short as 0.03 bpp (bit per pixel). Xiangwei et al. [32] stated that the LSB methods can result in the "pair effect" in the image histograms. As can be seen in Fig. 27, this "pair effect" phenomenon is empirically observed in steganography based on the modulus operator. Note that it is not always the case that modulus steganography produces such noticeable phenomenon. This operator acts as a means to generate random locations (i.e. not sequential) to embed data. It can be a complicated process or a simple

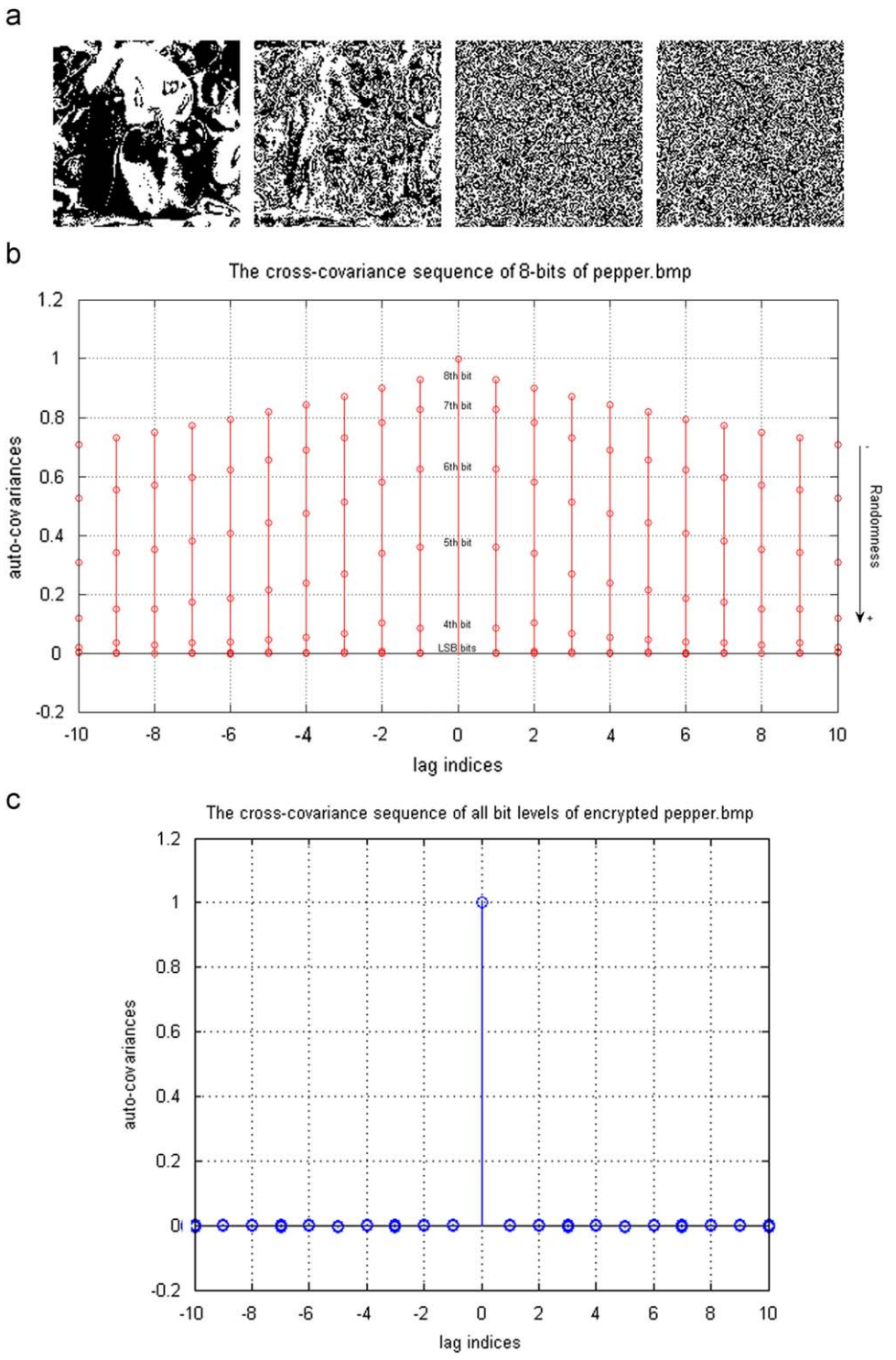

Fig. 23. Cross covariance test for randomness, (a) randomness in natural images, from left to right, original *pepper.bmp* 7th bit, 5th bit, 3rd bit and 2nd bit plan, respectively, (b) projection of each bit level from the plain image *pepper.bmp* and (c) a great randomness shown on all bit levels of the encrypted image. This phenomenon definitely helps mimic the least significant bits when embedding the encrypted secret data.

one like testing, in a raster scan fashion (if a pixel value is even then embed, otherwise do nothing). Avcibas et al. [106] applied binary similarity measures and multivariate regression to detect what they call "telltale marks" generated by the 7th and 8th bit planes of a stego-image.

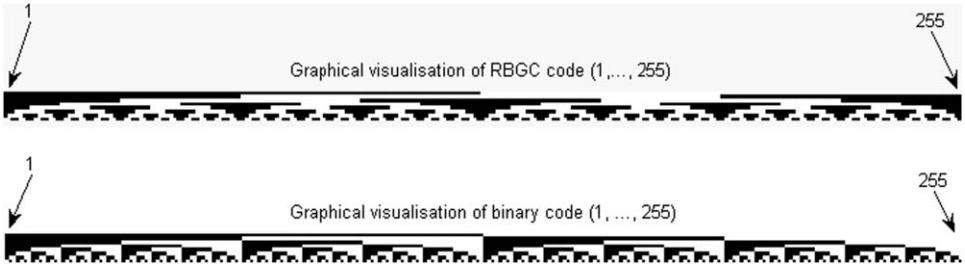

Fig. 24. RBGC and PBC (bottom) contrast in the graphical space.

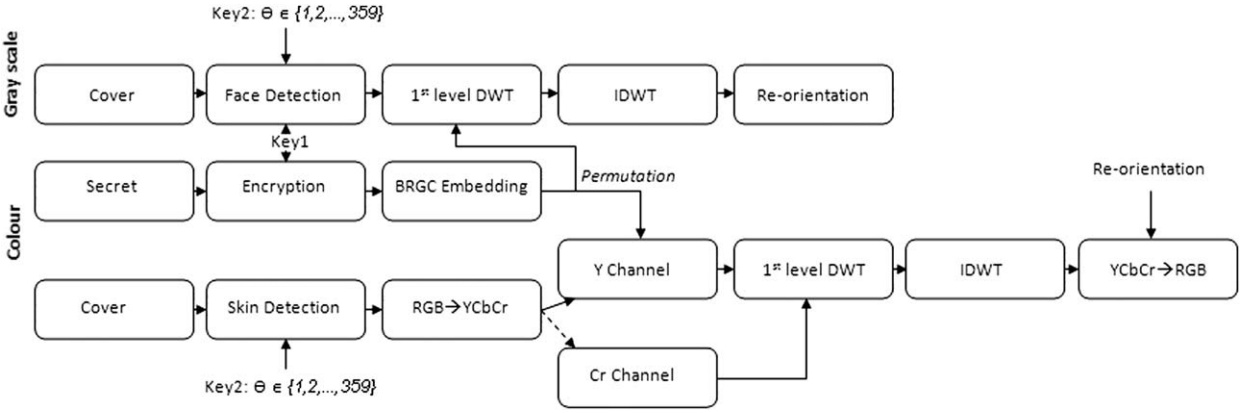

Fig. 25. Object based embedding introduced in [93]. Embedding into the "Y" channel has the advantage of better resistance to compression, while embedding into "Cr" channel has the advantage of better image perceptibility at the expense of resistance to image compression.

The previous histogram is given by the following discrete function:

$$H(k_i) = \sum_{i=0}^{255} g(k_i) \tag{9}$$

where k_i is the ith intensity level in the interval {0, 255} and $g(k_i)$ is the number of pixels in the image whose intensity level is k_i. It is the nature of standard intensity image histograms to track and graph frequencies of pixel values in a given image and not their structure and how they are arranged, see Fig. 28.

Chi-square (χ^2) and Pair-analysis algorithms can easily attack methods based on the spatial domain. Chi-square is a non-parametric (a rough estimate of confidence) statistical algorithm used in order to detect whether the intensity levels scatter in a uniform distribution throughout the image surface or not [107]. If one intensity level has been detected as such, then the pixels associated with this intensity level are considered as corrupted pixels or in this case have a higher probability of having embedded data. The classical Chi-square algorithm can be fooled by randomly embedded messages, thus Bohne and Westfeld [108] developed a steganalysis method to detect randomly scattered hidden data in the LSB spatial domain that applies the preserving statistical properties (PSP) algorithm.

If $o_i = \{o_1, o_2, \ldots, o_n\}$ denote the observed data, this can be seen as the number of times the symbols 1, 0 occur in the image LSBs [45, p. 311]. Let e_i be the number of times the event is expected to occur. Then the test statistic is of the form:

$$\chi^2 = \sum \frac{(o_i - e_i)^2}{e_i} \tag{10}$$

To avoid detection during steganalysis attacks, Fu and Au [109] and Guo (in watermarking) [110] proposed data hiding methods for halftone images. The assumption set here is that the inverse halftoning process would smooth the noise occurring from data embedding. However, inspired by the steganalysis techniques for gray level images, Cheng and Kot [111] successfully created a system able to counter-attack such methods by exploiting the wavelet statistic features extracted from the reconstructed gray level image through the inverse halftoning of a given halftone image fed into the support vector machine's classifier.

Jessica and Goljan [112] propose a statistical method that uses higher-order statistics called RS steganalysis; it is designed to provide an estimated percentage of flipped pixels caused by embedding as can be seen from Table 6 generated from Fig. 29.

Cancelli et al. [37] reveal that the performance of current state-of-the-art steganalysis algorithms for detection of ± 1 steganography is highly sensitive to the used training and testing databases. Their experiments also show that the examined algorithms are not applicable in their current state since the embedding rate for testing is very likely to be unknown, while it was assumed otherwise in those algorithms. Therefore, they conclude that no single steganalysis algorithm is constantly superior.

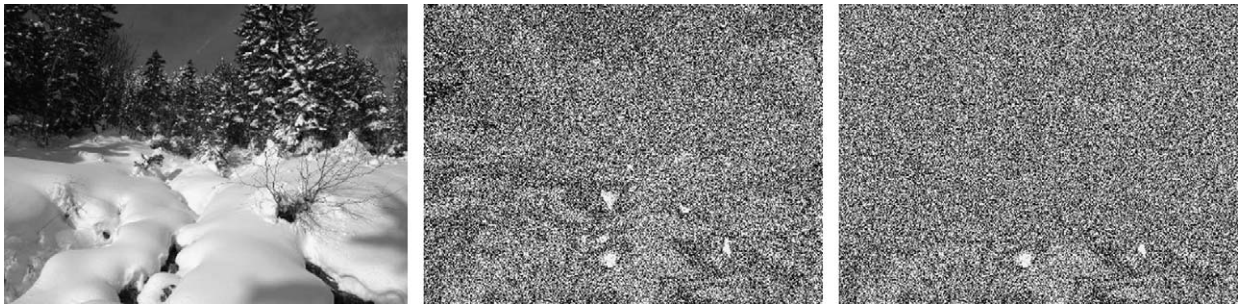

Fig. 26. Steganalysis using visual inspection: (left-to-right) original image, LSBs of the image before embedding and after embedding, respectively [104, pp. 16–17].

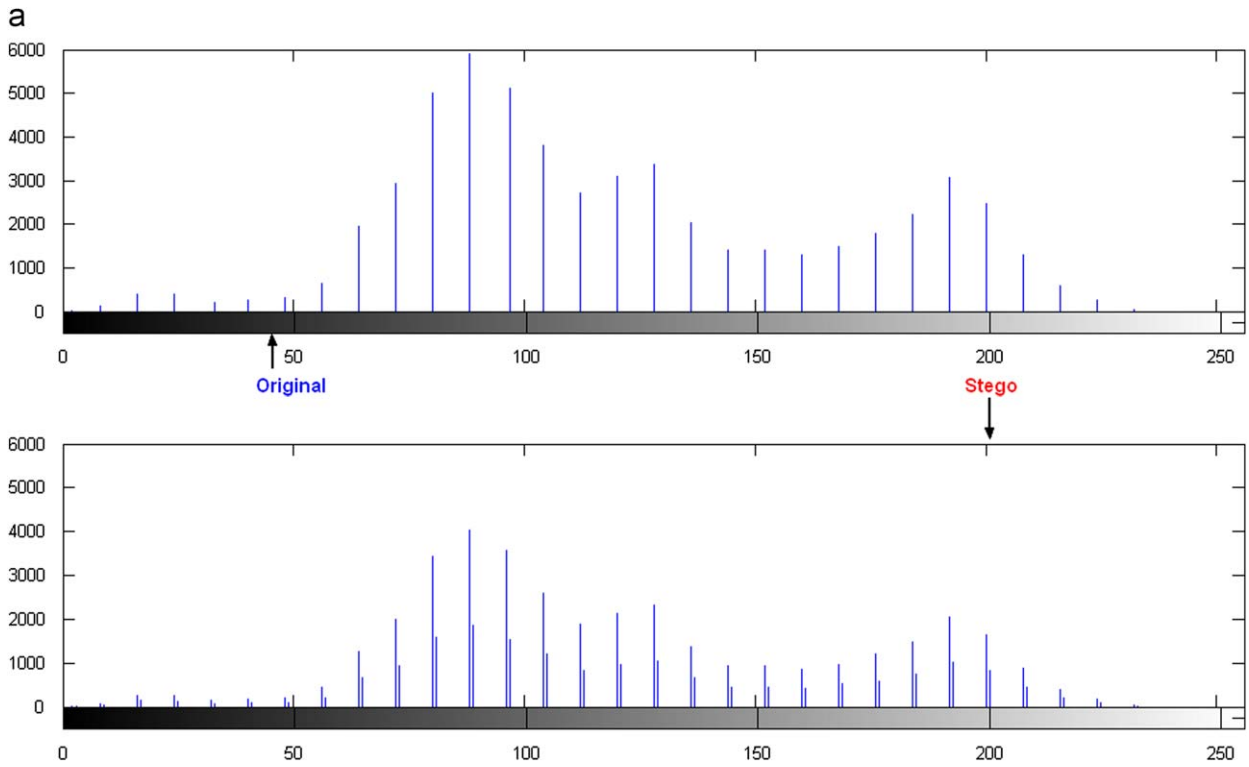

Fig. 27. Steganography based on modulus operators. Histograms demonstrating the "pair effect": (top) original and (bottom) stego-image.

In the frequency domain, Pevny and Jessica [113] developed a multi-class JPEG steganalysis system that comprised of DCT features and calibrated Markov features, which were then merged to produce a 274-dimensional feature vector. This vector is fed into a Support Vector Machine multi-classifier capable of detecting the presence of model-based steganography, F5, OutGuess, Steghide and JP Hide&Seek. Li et al. [114] exposed some of the weaknesses in "YASS" algorithm [115] by noticing that it introduces extra zero coefficients into the embedded host blocks because of the use of a quantization index modulation (QIM) method and by contrasting statistical features derived from different blocks in the stego-image.

Targeted embedding methods, such as the new enhanced MB2, are faced with much more accurate targeted attacks. That is because "if the selection channel is public, the attacker can focus on areas that were likely modified and use those less likely to have been modified for comparison/calibration purposes" [116, p. 6]. In [117], Ullerich and Westfeld, successfully attacked MB2 using coefficient types that are derived from the blockiness adjustment of MB2. They adapt Sallee's Cauchy model itself to detect Cauchy model-based embedded messages. In [118], Chen and Shi, attacked MB2 and other JPEG-based algorithms using Markov process (MP) that exploits the intra-block and inter-block correlations among JPEG coefficients.

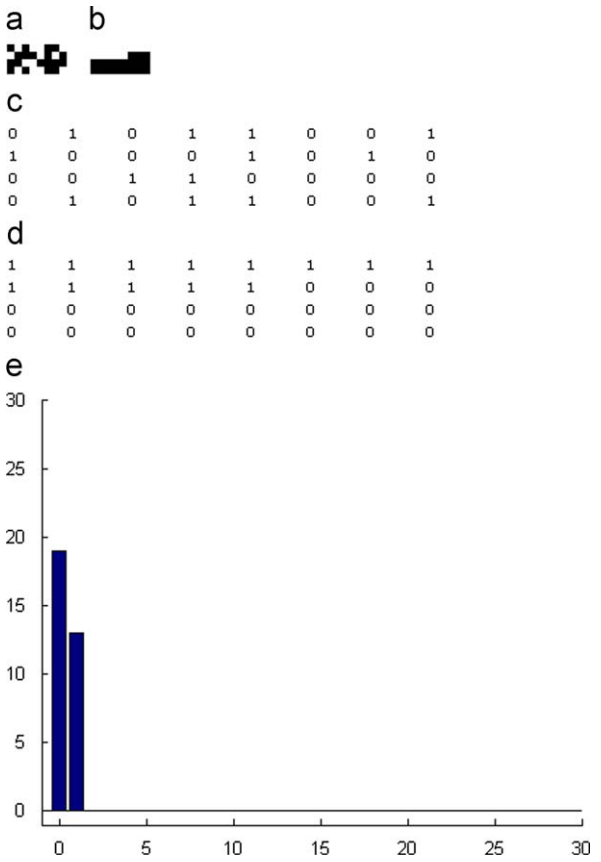

Fig. 28. Standard histogram is not meant for revealing the structure of data: (a) an 8×4 matrix stored in double precision and viewed, (b) another structure of (a), (c) pixel values of (a), (d) pixel values of (b) and (e) the histogram which describes both matrices.

Table 6
Estimated number of pixels with flipped LSBs for the test image in Fig. 29, with the actual numbers that should be detected in an ideal case (indicated in parenthesis) [112].

Image	Red (%)	Green (%)	Blue (%)
Cover image	2.5 (0.0)	2.4 (0.0)	2.6 (0.0)
Steganos	10.6 (9.8)	13.3 (9.9)	12.4 (9.8)
S-Tools	13.4 (10.2)	11.4 (10.2)	10.3 (10.2)
Hide4PGP	12.9 (10.0)	13.8 (10.1)	13.0 (10.0)

Fig. 29. An image used to test for the RS steganalysis' performance [112].

6. Conclusions and summary

This paper presented a background discussion on the major algorithms of steganography deployed in digital imaging. The emerging techniques such as DCT, DWT and adaptive steganography are not too prone to attacks, especially when the hidden message is small. This is because they alter coefficients in the transform domain, thus image distortion is kept to a minimum. Generally these methods tend to have a lower payload compared to spatial domain algorithms. There are different ways to reduce the bits needed to encode a hidden message. Apparent methods can be compression or correlated steganography, as proposed by Zheng and Cox [119], which is based on the conditional entropy of the message given the cover. In short, there has always been a trade-off between robustness and payload.

Scholars differ about the importance of robustness in steganography system design. In [120], Cox regards steganography as a process that should not consider robustness as it is then difficult to differentiate from watermarking. Katzenbeisser, on the other hand, dedicated a sub-section to robust steganography. He mentioned that robustness is a practical requirement for a steganography system. "Many steganography systems are designed to be robust against a specific class of mapping." [87, p. 32]. It is also rational to create an undetectable steganography algorithm that is capable of resisting common image processing manipulations that might occur by accident and not necessarily via an attack. Cox's view is formed based on his definition of steganography and its scope, while Katzenbeisser is looking at the process of steganography in a different way, preferring to view it as a robust secret communication mechanism.

Steganography urges that the cover image must be carefully selected. A familiar image should not be used, it is better for steganographers to create their own images [121]. This paper offered some guidelines and recommendations on the design of a steganographic system.

Steganography methods usually struggle with achieving a high embedding rate. As an alternative channel to images, video files have many excellent features for information hiding such as large capacity and good imperceptibility. The challenge, however, is to be able to embed into a group of images which are highly inter-correlated and often manipulated in a compressed form [122].

This paper also discusses with some detail the differences between steganography and watermarking. The various non-oblivious watermarking techniques available, which are highly resilient to image processing and geometric attacks, aim to detect the presence of a watermark using a correlation with an original template except in the rare watermarking blind detection scenario such as the work in [123]. This resilience can be seen for instance in the invariance proposed in the work of Deng et al. [124–126]. However, in steganography, this detection is not required as the aim is to correctly extract the hidden bits without the availability of any side information such as the original image and watermark.

Questions arise, such as whether child pornography exists inside seemingly innocent image or audio files? Are

criminals transmitting their secret messages in such a way? Are anti-virus systems fooled each time by secret embedding? The answers are still not trivial. However, what is evident is that steganography can have some useful applications, and like other technologies (i.e., encryption) it can be misused. These applications are numerous. For example, applying intelligent restricted content-based image retrieval (CBIR) [127], other avenues were highlighted in Section 2.

References

[1] N.F. Johnson, S. Jajodia, Exploring steganography: seeing the unseen, IEEE Computer 31 (2) (1998) 26–34.

[2] J.C. Judge, Steganography: past, present, future. SANS Institute publication, 〈http://www.sans.org/reading_room/whitepapers/stenganography/552.php〉, 2001.

[3] N. Provos, P. Honeyman, Hide and seek: an introduction to steganography, IEEE Security and Privacy 1 (3) (2003) 32–44.

[4] P. Moulin, R. Koetter, Data-hiding codes, Proceedings of the IEEE 93 (12) (2005) 2083–2126.

[5] S.B. Sadkhan, Cryptography: current status and future trends, in: Proceedings of IEEE International Conference on Information & Communication Technologies: From Theory to Applications, Damascus. Syria, April 19–23, 2004, pp. 417–418.

[6] S. Lyu, H. Farid, Steganalysis using higher-order image statistics, IEEE Transactions on Information Forensics and Security 1 (1) (2006) 111–119.

[7] D. Kahn, The codebreakers: the comprehensive history of secret communication from ancient times to the Internet, Scribner, December 5, 1996.

[8] J.P. Delahaye, Information noyée, information cach, Pour la Science 229 (1996) 142–146 〈www.apprendre-en-ligne.net/crypto/steganо/229_142_146.pdf〉 (in French)..

[9] G.J. Simmons, The prisoners' problem and the subliminal channel, in: Proceedings of International Conference on Advances in Cryptology, CRYPTO83, August 22–24, 1984, pp. 51–67.

[10] C. Kurak, J. McHugh, A cautionary note on image downgrading, in: Proceedings of the IEEE 8th Annual Computer Security Applications Conference, 30 November–4 December, 1992, pp. 153–159.

[11] T.L. Thomas, Al qaeda and the internet: the danger of "cyberplanning", parameters, US Army War College Quarterly-Spring 2003. Available from: 〈www.carlisle.army.mil/usawc/Parameters/03spring/thomas.pdf〉.

[12] C. Hosmer, Discovering hidden evidence, Journal of Digital Forensic Practice (1) (2006) 47–56.

[13] J.C. Hernandez-Castro, I. Blasco-Lopez, J.M. Estevez-Tapiador, Steganography in games: a general methodology and its application to the game of Go, Computers and Security, Elsevier Science 25 (2006) 64–71.

[14] P. Hayati, V. Potdar, E. Chang, A survey of steganographic and steganalytic tools for the digital forensic investigator, available from: 〈http://debii.curtin.edu.au/~pedram/images/docs/survey_of_steganography_and_steganalytic_tools.pdf〉.

[15] W. Bender, W. Butera, D. Gruhl, R. Hwang, F.J. Paiz, S. Pogreb, Applications for data hiding, IBM Systems Journal 39 (3&4) (2000) 547–568.

[16] F.A.P. Petitcolas, Introduction to information hiding, in: S. Katzenbeisser, F.A.P. Petitcolas (Eds.), Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Inc., Norwood, 2000.

[17] S. Miaou, C. Hsu, Y. Tsai, H. Chao, A secure data hiding technique with heterogeneous data-combining capability for electronic patient records, in: Proceedings of the IEEE 22nd Annual EMBS International Conference, Chicago, USA, July 23–28, 2000, pp. 280–283.

[18] U.C. Nirinjan, D. Anand, Watermarking medical images with patient information, in: Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Hong Kong, China, 29 October–1 November 1998, pp. 703–706.

[19] Y. Li, C. Li, C. Wei, Protection of mammograms using blind steganography and watermarking, in: Proceedings of the IEEE International Symposium on Information Assurance and Security, 2007, pp. 496–499.

[20] D. Frith, Steganography approaches, options, and implications, Network Security 2007 (8) (2007) 4–7.

[21] H. Farid, A survey of image forgery detection, IEEE Signal Processing Magazine 26 (2) (2009) 16–25.

[22] A. Cheddad, J. Condell, K. Curran, P. Mc Kevitt, A secure and improved self-embedding algorithm to combat digital document forgery, Signal Processing 89 (12) (2009) 2324–2332.

[23] N.F. Johnson, S.C. Katzenbeisser, A survey of steganographic techniques, in: S. Katzenbeisser, F.A.P. Petitcolas (Eds.), Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Inc., Norwood, 2000.

[24] K. Bailey, K. Curran, An evaluation of image based steganography methods, Multimedia Tools and Applications 30 (1) (2006) 55–88.

[25] [Hide and Seek]: 〈ftp://ftp.funet.fi/pub/crypt/mirrors/idea.sec.dsi.unimi.it/cypherpunks/steganography/hdsk41b.zip〉;
[S-Tools]: 〈ftp://ftp.funet.fi/pub/crypt/mirrors/idea.sec.dsi.unimi.it/code/s-tools4.zip〉;
[Stella]: 〈http://wwwicg.informatik.uni-rostock.de/~sanction/stella/〉;
[Hide in Picture]: 〈http://sourceforge.net/projects/hide-in-picture/〉;
[Revelation]: 〈http://revelation.atspace.biz/〉;
[Camouflage]: 〈http://camouflage.unfiction.com/〉;
[JpegX]: 〈http://www.freewarefiles.com/Jpegx_program_19392.html〉;
[Data Stash]: 〈http://www.skyjuicesoftware.com/software/ds_info.html〉;
[Other Tools]: 〈http://www.jjtc.com/Security/stegtools.htm〉;
[F5]: 〈http://wwwrn.inf.tu-dresden.de/~westfeld/f5.html〉;
[OutGuess]: 〈http://www.outguess.org/〉.

[26] P. Alvarez, Using extended file information (EXIF) file headers in digital evidence analysis, International Journal of Digital Evidence, Economic Crime Institute (ECI) 2 (3) (2004) 1–5.

[27] V.M. Potdar, S. Han, E. Chang, Fingerprinted secret sharing steganography for robustness against image cropping attacks, in: Proceedings of IEEE Third International Conference on Industrial Informatics (INDIN), Perth, Australia, 10–12 August 2005, pp. 717–724.

[28] M.H. Shirali-Shahreza, M. Shirali-Shahreza, A new approach to Persian/Arabic text steganography, in: Proceedings of Fifth IEEE/ACIS International Conference on Computer and Information Science (ICIS-COMSAR 2006), 10–12 July 2006, pp. 310–315.

[29] E.T. Lin, E.J. Delp, A review of data hiding in digital images, in: Proceedings of the Image Processing, Image Quality, Image Capture Systems Conference, PICS'99, the Society for Imaging Science and Technology, 1999, pp. 274–278.

[30] C.C. Chang, C.Y. Lin, Y.Z. Wang, New image steganographic methods using run-length approach, Information Sciences 176 (22) (2006) 3393–3408.

[31] R.J. Hwang, K.T. Shih, C.H. Kao, T.M. Chang, Lossy compression tolerant steganography, in: Proceedings of the First International Conference on The Human Society and the Internet–Internet Related Socio-Economic Issues, Lecture Notes in Computer Science, 2001, vol. 2105, pp. 427–435.

[32] X. Kong, Z. Wang, X. You, Steganalysis of palette images: attack optimal parity assignment algorithm, in: Proceedings of Fifth IEEE International Conference on Information, Communications and Signal Processing, 06–09 December 2005, pp. 860–864.

[33] J. Fridrich, M. Goljan, D. Hogeg, Steganalysis of JPEG images: breaking the F5 algorithm, in: Proceedings of Information Hiding: Fifth International Workshop, IH 2002 Noordwijkerhout, The Netherlands, Lecture Notes in Computer Science, Springer, October 7–9, 2002, 2578/2003, pp. 310–323.

[34] K.H. Jung, K.Y. Yoo, Data hiding method using image interpolation, Computer Standards and Interfaces 31 (2) (2009) 465–470.

[35] Z. Li, X. Chen, X. Pan, X. Zeng, Lossless data hiding scheme based on adjacent pixel difference, in: Proceedings of the International Conference on Computer Engineering and Technology, 2009, pp. 588–592.

[36] P. Tsai, Y.C. Hu, H.L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, Signal Processing 89 (6) (2009) 1129–1143.

[37] G. Cancelli, G.J. Doërr, M. Barni, I.J. Cox, A comparative study of ± 1 steganalyzers, in: Proceedings of IEEE 10th Workshop on Multimedia Signal Processing, MMSP'08, 8–10 October 2008, pp. 791–796.

[38] A.C. Popescu, Statistical tools for digital image forensics, Ph.D. Dissertation, Department of Computer Science, Dartmouth College, USA, 2005. Available from: 〈http://www.cs.dartmouth.edu/~farid/publications/apthesis05.html〉, on 16-05-07 at 12:20.

[39] X. Li, J. Wang, A steganographic method based upon JPEG and particle swarm optimization algorithm, Information Sciences 177 (15) (2007) 3099–31091.

[40] C.C. Chang, T.S. Chen, L.Z. Chung, A steganographic method based upon JPEG and quantization table modification, Information Sciences 141 (1–2) (2002) 123–138.
[41] A.M. Fard, M. Akbarzadeh-T, F. Varasteh-A, A new genetic algorithm approach for secure JPEG steganography, in: Proceedings of IEEE International Conference on Engineering of Intelligent Systems, 22–23 April 2006, pp. 1–6.
[42] A.I. Hashad, A.S. Madani, A.E.M.A. Wahdan, A robust steganography technique using discrete cosine transform insertion, in: Proceedings of IEEE/ITI Third International Conference on Information and Communications Technology, Enabling Technologies for the New Knowledge Society, 5–6 December 2005, pp. 255–264.
[43] K.B. Raja, C.R. Chowdary, K.R. Venugopal, L.M. Patnaik, A secure image steganography using LSB, DCT and compression techniques on raw images, in: Proceedings of IEEE 3rd International Conference on Intelligent Sensing and Information Processing, ICISIP'05, Bangalore, India, 14–17 December 2005, pp. 170–176.
[44] R.T. McKeon, Strange Fourier steganography in movies, in: Proceedings of the IEEE International Conference on Electro/Information Technology (EIT), 17–20 May 2007, pp. 178–182.
[45] P. Wayner, Disappearing Cryptography, second ed, Morgan Kaufmann Publishers, 2002.
[46] C. Manikopoulos, S. Yun-Qing, S. Sui, Z. Zheng, N. Zhicheng, Z. Dekun, Detection of block DCT-based steganography in gray-scale images, in: Proceedings of the IEEE Workshop on Multimedia Signal Processing, 9–11 December 2002, pp. 355–358.
[47] N. Provos, Defending against statistical steganalysis, Center for Information Technology Integration, University of Michigan, Technical report, February 2001.
[48] N. Provos, P. Honeyman, Detecting steganographic content on the Internet, Center for Information Technology Integration, University of Michigan, Technical report, August 31, 2001.
[49] A. Westfeld, F5-A steganographic algorithm: high capacity despite better steganalysis, in: Proceedings of Fourth International Workshop on Information Hiding, Lecture Notes in Computer Science, vol. 2137, Pittsburgh, USA, April 2001, pp. 289–302.
[50] J. Fridrich, T. Pevny, J. Kodovsky, Statistically undetectable JPEG steganography: dead ends, challenges, and opportunities, in: Proceedings of the ACM Ninth Workshop on Multimedia & Security, Dallas, Texas, USA, September 20–21, 2007, pp. 3–14.
[51] J. Fridrich, M. Goljan, D. Soukal, Perturbed quantization steganography, ACM Multimedia and Security Journal 11 (2) (2005) 98–107.
[52] W.Y. Chen, Color image steganography scheme using set partitioning in hierarchical trees coding, digital Fourier transform and adaptive phase modulation, Applied Mathematics and Computation 185 (1) (2007) 432–448.
[53] V.M. Potdar, S. Han, E. Chang, A survey of digital image watermarking techniques, in: Proceedings of the IEEE Third International Conference on Industrial Informatics (INDIN), Perth, Australia, 10–12 August 2005, pp. 709–716.
[54] B. Verma, S. Jain, D.P. Agarwal, Watermarking image databases: a review, in: Proceedings of the International Conference on Cognition and Recognition, Mandya, Karnataka, India, 22–23 December 2005, pp. 171–179.
[55] N.K. Abdulaziz, K.K. Pang, Robust data hiding for images, in: Proceedings of IEEE International Conference on Communication Technology, WCC-ICCT'02, vol. 1, 21–25 August 2000, pp. 380–383.
[56] L.D. Paulson, New system fights steganography, News Briefs, IEEE Computer Society 39 (8) (2006) 25–27.
[57] A.A. Abdelwahab, L.A. Hassan, A discrete wavelet transform based technique for image data hiding, in: Proceedings of 25th National Radio Science Conference, NRSC 2008, Egypt, March 18–20, 2008, pp. 1–9.
[58] P. Sallee, Model-based steganography, in: Proceedings of the Second International Workshop on Digital Watermarking, Seoul, Korea, October 20–22, 2003, Lecture Notes in Computer Science, vol. 2939, pp. 254–260.
[59] M. Kharrazi, H.T. Sencar, N. Memon, Performance study of common image steganography and steganalysis techniques, Journal of Electrical Imaging 15 (4) (2006) 1–16.
[60] R. Tzschoppe, R. Baum, J. Huber, A. Kaup, Steganographic system based on higher-order statistics, in: Proceedings of SPIE, Security and Watermarking of Multimedia Contents V. Santa Clara, California, USA 2003, vol. 5020, pp. 156–166.
[61] C.C. Chang, H.W. Tseng, A steganographic method for digital images using side match, Pattern Recognition Letters 25 (12) (2004) 1431–1437.
[62] E. Franz, A. Schneidewind, Adaptive steganography based on dithering, in: Proceedings of the ACM Workshop on Multimedia and Security, September 20–21, 2004, Magdeburg, Germany, pp. 56–62.
[63] R. Böhme, A. Westfeld, Breaking cauchy model-based JPEG steganography with first order statistics, in: Proceedings of the European Symposium on Research in Computer Security, ESORICS 2004, Valbonne, France, 13th September 2004, Lecture Notes in Computer Science, vol. 3193, pp. 125–140.
[64] L. Yu, Y. Zhao, R. Ni, Z. Zhu, PM1 steganography in JPEG images using genetic algorithm, Soft Computing 13 (4) (2009) 393–400.
[65] C.C. Chang, P. Tsai, M.H. Lin, An adaptive steganography for index-based images using codeword grouping, Advances in Multimedia Information Processing-PCM, Springer, vol. 3333, 2004, pp. 731–738.
[66] H. Hioki, A data embedding method using BPCS principle with new complexity measures, in: Proceedings of Pacific Rim Workshop on Digital Steganography, July 2002, pp. 30–47.
[67] K.B. Raja, S. Sindhu, T.D. Mahalakshmi, S. Akshatha, B.K. Nithin, M. Sarvajith, K.R. Venugopal, L.M. Patnaik, Robust image adaptive steganography using integer wavelets, in: Proceedings of the Third International Conference on Communication Systems Software and Middleware and Workshops, COMSWARE'08, 6–10 January 2008, pp. 614–621.
[68] Y. Srinivasan, High capacity data hiding system using BPCS steganography, Master Dissertation, Texas Tech. University, USA, December 2003, pp. 8, available from: 〈http://etd.lib.ttu.edu/theses/available/etd-06272008-31295018922590/unrestricted/31295018922590.pdf〉.
[69] J. Spaulding, H. Noda, M.N. Shirazi, E. Kawaguchi, BPCS steganography using EZW lossy compressed images, Pattern Recognition Letters 23 (13) (2002) 1579–1587.
[70] J. Fridrich, Application of data hiding in digital images, Tutorial for the ISSPA'99, Brisbane, Australia, August 22–25, 1999.
[71] Y. Srinivasan, B. Nutter, S. Mitra, B. Phillips, D. Ferris, Secure transmission of medical records using high capacity steganography, in: Proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems, CBMS'04, 2004, pp. 122–127.
[72] E. Kawaguchi, R.O. Eason, Principle and applications of BPCS steganography, in: Proceedings of SPIE International Symposium on Voice, Video, and Data Communications, 2–4 November 1998, pp. 464–473.
[73] C.C. Lin, W.L. Tai, C.C. Chang, Multilevel reversible data hiding based on histogram modification of difference images, Pattern Recognition 41 (12) (2008) 3582–3591.
[74] Y.T. Wu, F.Y. Shih, Genetic algorithm based methodology for breaking the steganalytic systems, IEEE Transactions on Systems, Man, and Cybernetics—part B: cybernetics 36 (1) (2006) 24–31.
[75] S.P. Maity, M.K. Kundu, P.K. Nandi, Genetic algorithm for optimal imperceptibility in image communication through noisy Channel, in: Proceedings of the International Conference on Neural Information Processing (ICONIP '2004), India, 29 October 2004, pp. 700–705.
[76] J. Kong, H. Jia, X. Li, Z. Qi, A novel content-based information hiding scheme, in: Proceedings of the International Conference on Computer Engineering and Technology, 22–24 January 2009, vol. 1, pp. 436–440.
[77] M.W. Chao, C.H. Lin, C.W. Yu, T.Y. Lee, A high capacity 3D steganography algorithm, IEEE Transactions on Visualization and Computer Graphics 15 (2) (2009) 274–284.
[78] A. Bogomjakov, C. Gotsman, M. Isenburg, Distortion-free steganography for polygon meshes, in: Proceedings of Computer Graphics Forum, Eurographics'08, April 2008, vol. 27 (2), pp. 637–642.
[79] H. Nakamura, Q. Zhao, Information hiding based on image morphing, in: Proceedings of 22nd International Conference on Advanced Information Networking and Applications Workshops, AINAW, 25–28 March 2008, pp. 1585–1590.
[80] A.M. Zeki, A.A. Manaf, A novel digital watermarking technique based on ISB (Intermediate Significant Bit), World Academy of Science, Engineering and Technology 38 (2009) 1080–1087.
[81] Y.H. Yu, C.C. Chang, I.C. Lin, A new steganographic method for color and grayscale image hiding, Computer Vision and Image Understanding 107 (3) (2007) 183–194.
[82] M. Drew, S. Bergner, Spatio-chromatic decorrelation for color image compression, Technical Report, School of Computing Science, Simon Fraser University, Vancouver, Canada, 2007, available from: 〈http://fas.sfu.ca/pub/cs/TR/2007/CMPT2007-09.pdf〉.
[83] M. Saenz, R. Oktem, K. Egiazarian, E. Delp, Color image wavelet compression using vector morphology, in: Proceedings of the

European Signal Processing Conference, September 5–8 2000, Tampere, Finland, 2000, pp. 5–8.
[84] A. Rodriguez, L. Rowe, Multimedia systems and applications, IEEE Computer 28 (5) (1995) 20–22.
[85] D. Van Der Weken, M. Nachtegael, E. Kerre, Using similarity measures and homogeneity for the comparison of images, Image and Vision Computing 22 (9) (2004) 695–702.
[86] M. Kutter, F. Petitcolas, A fair benchmark for image watermarking systems, in: Proceedings of Electronic Imaging '99, Security and Watermarking of Multimedia Contents, San Jose, California, USA, 25–27 January 1999, vol. 3657, pp. 226–239.
[87] S.C. Katzenbeisser, Principles of steganography, in: S. Katzenbeisser, F.A.P. Petitcolas (Eds.), Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Inc, Norwood, 2000.
[88] J. Fridrich, M. Goljan, Practical steganalysis of digital images-state of the art, in: Proceedings of SPIE Photonics West, Electronic Imaging'02, Security and Watermarking of Multimedia Contents, San Jose, California, January 2002, vol. 4675, pp. 1–13.
[89] A. Martin, G. Sapiro, G. Seroussi, Is image steganography natural?, IEEE Transactions on Image Processing 14 (12) (2005) 2040–2050.
[90] S. Areepongsa, N. Kaewkamnerd, Y.F. Syed, K.R. Rao, Exploring on steganography for low bit rate wavelet based coder in image retrieval system, in: Proceedings of IEEE TENCON, Kuala Lumpur, Malaysia, 2000, vol. 3, pp. 250–255.
[91] P. Kruus, C. Scace, M. Heyman, M. Mundy, A survey of steganographic techniques for image files, Advanced Security Research Journal V (I) (2003) 41–51.
[92] A. Cheddad, J. Condell, K. Curran, P. Mc Kevitt, Skin tone based steganography in video files exploiting the YCbCr colour space, in: Proceedings of the IEEE International Conference on Multimedia and Expo, Hannover, Germany, June 23–26, 2008, pp. 905–909.
[93] A. Cheddad, J. Condell, K. Curran, P. Mc Kevitt, A skin tone detection algorithm for an adaptive approach to steganography, Signal Processing 89 (12) (2009) 2465–2478.
[94] A. Nikolaidis, I. Pitas, Region-based image watermarking, IEEE Transactions on Image Processing 10 (11) (2001) 1726–1740.
[95] A. Nikolaidis, I. Pitas, Robust watermarking of facial images based on salient geometric pattern matching, IEEE Transactions on Multimedia 2 (3) (2000) 172–184.
[96] D.C. Lou, C.H. Sung, A steganographic scheme for secure communications based on the chaos and Euler theorem, IEEE Transactions on Multimedia 6 (3) (2004) 501–509.
[97] A. Cheddad, J. Condell, K. Curran, P. Mc Kevitt, Securing information content using new encryption method and steganography, in: Proceedings of the Third IEEE International Conference on Digital Information Management, University of East London, UK, 13–16 November 2008, pp. 563–568.
[98] D.C. Wu, W.H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recognition Letters 24 (9–10) (2003) 1613–1626.
[99] J. Kodovsky, J. Fridrich, Influence of embedding strategies on security of steganographic methods in the JPEG domain, in: Proceedings of SPIE Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, San Jose, CA, January 28–30, 2008, vol. 6819, pp. 1–13.
[100] Y.S. Chen, R.Z. Wang, Steganalysis of reversible contrast mapping watermarking, IEEE Signal Processing Letters 16 (2) (2009) 125–128.
[101] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, A statistical test suite for random and pseudorandom number generators for cryptographic applications, National Institute of Standards and Technology (NIST), special publication 800-22, August 2008.
[102] L.M. Marvel, C.T. Retter, A methodology for data hiding using images, in: Proceedings of IEEE Military Communications Conference, MILCOM'98, Boston, MA, USA, 18–21 October 1998, pp. 1044–1047.
[103] R.J. Anderson, F.A.P. Petitcolas, On the limits of steganography, IEEE Journal of Selected Areas in Communications 16 (4) (1998) 474–481.
[104] P. Bas, Analyse stéganographique d'images numériques: Comparaison de différentes méthodes, Rapport de stage, Laboratoire des Images et des Signaux, University of Joseph Fourier, 23rd June 2003, (in French).
[105] J. Fridrich, M. Goljan, R. Du, Reliable detection of LSB steganography in grayscale and color images, in: Proceedings of ACM, Special Session on Multimedia Security and Watermarking, Ottawa, Canada, 5th October 2001, pp. 27–30.
[106] I. Avcibas, N. Memon, B. Sankur, Image steganalysis with binary similarity measures, in: Proceedings of the IEEE International Conference on Image Processing, 24–28 June 2002, vol. 3, pp. 645–648.
[107] P. Civicioglu, M. Alci, E. Besdok, Impulsive noise suppression from images with the noise exclusive filter, EURASIP Journal on Applied Signal Processing 2004 (16) (2004) 2434–2440.
[108] R. Bohme, A. Westfeld, Exploiting preserved statistics for steganalysis, Lecture Notes in Computer Science, vol. 3200/2005, Springer, Berlin, 2005, pp. 82–96.
[109] M.S. Fu, O.C. Au, Data hiding watermarking for halftone images, IEEE Transactions on Image Processing 11 (4) (2002) 477–484.
[110] J.M. Guo, Watermarking in dithered halftone images with embeddable cells selection and inverse halftoning, Signal Processing 88 (6) (2008) 1496–1510.
[111] J. Cheng, A.C. Kot, Steganalysis of halftone image using inverse halftoning, Signal Processing 89 (6) (2009) 1000–1010.
[112] J. Fridrich, M. Goljan, R. Du, Detecting LSB steganography in color and gray-scale images, IEEE Multimedia 8 (4) (2001) 22–28.
[113] T. Pevny, J. Fridrich, Merging Markov and DCT features for multi-class JPEG steganalysis, in: Proceedings of SPIE Electronic Imaging, Photonics West, USA, January 2007, pp. 03–04.
[114] B. Li, Y.Q. Shi, J. Huang, Steganalysis of YASS, in: Proceedings of 10th ACM Workshop on Multimedia and Security, Oxford, United Kingdom, 22–23 September 2008, pp. 139–148.
[115] K. Solanki, A. Sarkar, B.S. Manjunath, YASS: yet another steganographic scheme that resists blind steganalysis, in: Proceedings of the Ninth International Workshop on Information Hiding, Saint Malo, France, 11–13 June 2007, Lecture Notes in Computer Science, vol. 4567, pp. 16–31.
[116] J. Kodovský, J. Fridrich, Influence of embedding strategies on security of steganographic methods in the JPEG domain, in: Proceedings of SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, San Jose, CA, January 28–31, 2008, vol. 6819, pp. 1–13.
[117] C. Ullerich, A. Westfeld, Weaknesses of MB2, in: Proceedings of the Sixth International Workshop on Digital Watermarking, Guangzhou, China, December 3–5, 2007, pp. 127–142.
[118] C. Chen, Y.Q. Shi, JPEG image steganalysis utilizing both intrablock and interblock correlations, in: Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS 2008, Seattle, Washington, USA, 18–21 May 2008, pp. 3029–3032.
[119] L. Zheng, I. Cox, JPEG based conditional entropy coding for correlated steganography, in: Proceedings of IEEE International Conference on Multimedia and Expo, Beijing, China, 2–5 July 2007, pp. 1251–1254.
[120] I. Cox, Information hiding, watermarking and steganography, Public Seminar, Intelligent Systems Research Centre (ISRC), University of Ulster at Magee, Northern Ireland, 28th April 2009.
[121] K. Curran, X. Li, R. Clarke, An investigation into the use of the least significant bit substitution technique in digital watermarking, American Journal Applied Sciences 2 (3) (2005) 648–654.
[122] Z. Zhao, N. Yu, X. Li, A novel video watermarking scheme in compression domain based on fast motion estimation, in: Proceedings of IEEE International Conference on Communication Technology, 2003, pp. 1878–1882.
[123] C. Deng, X. Gao, X. Li, D. Tao, A local Tchebichef moments-based robust image watermarking, Signal Processing 89 (8) (2009) 1531–1539.
[124] C. Deng, X. Gao, D. Tao, X. Li, Digital watermarking in image affine co-variant regions, in: International Conference on Machine Learning and Cybernetics, vol. 4, 2007, pp. 2125–2130.
[125] C. Deng, X. Gao, D. Tao, X. Li, Geometrically invariant watermarking using affine covariant regions, in: Proceedings of IEEE International Conference on Image Processing, 2008, pp. 413–416.
[126] C. Deng, X. Gao, D. Tao, X. Li, Invariant image watermarking based on local feature regions, in: Proceedings of International Conference on Cyberworlds, 2008, pp. 6–10.
[127] X. Li, Watermarking in secure image retrieval, Pattern Recognition Letters 24 (14) (2003) 2431–2434.