Noname manuscript No.
(will be inserted by the editor)

Maximizing Embedding Capacity *and* Stego Quality: Curve-Fitting in the Transform Domain

Tamer Rabie · Ibrahim Kamel · Mohammed Baziyad

Received: date / Accepted: date

Abstract Achieving high embedding capacities for information hiding systems while maintaining high perceptual stego quality is a critical challenge in steganography. This quandary is attracting researchers to overcome the trade-off barrier between high capacities and enhanced levels of stego image quality. This work introduces a promising transform-domain hiding scheme that aims to achieve ultimate hiding capacity with premium perceptual quality results. The proposed scheme is based on the fact that highly correlated images are represented by significant coefficients that are strongly packed in the transform-domain of the image. This allows for a large space in the insignificant coefficient areas to embed in. To exploit this feature optimally, a curve-fitting approach is introduced and implemented in various adaptive-region transform-domain embedding schemes. Experimental results demonstrate that this curve-fitting methodology is able to enhance adaptive transform-domain embedding schemes where very high embedding capacities can be achieved that are much higher than competing high-capacity hiding schemes. The other noticeable re-

T. Rabie
Associate Professor
Department of Electrical and Computer Engineering
University of Sharjah, U.A.E.
Phone: +97165053943
E-mail: trabie@sharjah.ac.ae

I. Kamel
Professor
Department of Electrical and Computer Engineering
University of Sharjah, U.A.E.
E-mail: kamel@sharjah.ac.ae

M.Baziyad
Research Assistant
Research Institute of Sciences & Engineering
University of Sharjah, U.A.E.
E-mail: mbaziyad@sharjah.ac.ae

sult is that although the embedding capacity has increased compared to earlier work, the perceptual quality level has also improved over previous methods.

Keywords

Curve Fitting; High Capacity; High Perceptual Quality; Steganography; Transform-domain Image Hiding; Quad-Tree; Discrete cosine transform; Discrete Wavelet Transform

1 INTRODUCTION

Steganography is the art that deals with concealing the existence of messages. The invention of this art is not new, some historical studies traced back this science to 440BC [17]. However, the interest in information security techniques, such as steganography and encryption, has widely grown in recent decades with the massive increase in the exchanged multimedia data over insecure networks [6, 40, 14, 20, 21].

Steganography differs from encryption in that encryption is concerned with encoding messages, so that only authorized entities can understand them. Steganography adds a further security step since it conceals the occurrence of the hidden data exchanged. "Watermarking", on the other hand, is a different category of data hiding techniques. This class is used for authentication and integrity purposes [23].

There are four different aspects that researchers try to improve; capacity, perceptibility, robustness, and security. The capacity attribute refers to the size of the hidden data in a cover medium. Perceptibility deals with the amount of "noise" in the cover medium. Robustness refers to "solidity" of the stego medium, and its ability to keep the secret information undestroyed with the existence of noise and impairments, and security refers to an eavesdropper's inability to detect and inturn extract or change the hidden information.

The recent growth in data sizes has attracted many researchers to contribute in the embedding capacity area of research [2, 16, 8, 9, 5, 18, 30, 25]. The drive to improve the embedding capacity is hindered by the fact that an increase in the amount of embedded secret data will typically result in poor stego channel quality.

In this work, we present a transform-domain embedding scheme that addresses the shortcomings in previous high capacity embedding schemes where researchers had to trade-off between higher capacities and reduced perceptual quality or choose higher perceptual quality albeit at the expense of lower capacities.

In images, the core concept behind data embedding is the fact that most images can be partitioned into different regions based on their inter-pixel relations. These regions vary between high frequency areas where pixels in these areas have almost no correlation, and low frequency areas (highly correlated). Embedding schemes will usually find the redundancy in the pixel information

of the cover image where the correlation is at its minimum level, and use it to hide the secret data.

Recent work by Rabie & Kamel [33,34] has tried to address the problem of capacity versus perceptual stego quality by estimating adaptive *square regions* inside fixed and adaptive blocks of the DCT of the cover image. These schemes were able to reach extremely high embedding capacities above 21bpp while maintaining perceptibility at an acceptable level of around 27dB.

The steganography scheme described in this paper is mainly based on the idea that the DCT coefficients of a correlated image will be strongly packed in the top left region of the DCT domain. This suggests segmenting the cover image into different segments based on their inter-pixel correlation level. By transforming each segment using DCT, we insure having a very large area to embed in, as important coefficients will be concentrated in a small area due to the strong "energy compaction" property of the DCT.

Unlike previous adaptive region embedding approaches [34,33], where only a *square region* in the lower-right corner of the possible hiding area in the DCT block is used to embed in, our new scheme introduces a *curve-fitting* (CF) approach to utilize fully the whole area that is suitable to embed in. This has led to a substantial increase in the available embedding area in the DCT of the cover image and has consequently resulted in improved hiding capacities while achieving very high perceptibility values for the stego image in comparison to these earlier schemes. As a matter of fact, we were able to reach an embedding capapcity of 22bpp at PSNR 35.83dB, as will be shown in detail in section 5.

The rest of this paper is organized as follows. In section 2 we present prior work in the area of high-capacity data hiding. Section 3 briefly reviews the theory of the DCT transform and its wide use in image compression. Our proposed Curve-Fitting methodology implemeted in the Quad-Tree Adaptive Region scheme of [34], which we denote as (CF-QTAR), is discussed in section 4. Section 5 presents our comparison results and demonstrates the highest capacity/perceptibility levels that can be achieved based on our approach. Finally, concluding remarks appear in section 6.

2 Related Work

The traditional methods to embed information into a cover image are the Least Significant Bit (LSB) methods. The general idea of this technique is to embed in the least significant bit in each pixel of the cover image. There are many implementations of this method. In [39] only one of the three channels at each pixel of the cover image is elected by a Sample Pairs analysis, then a LSB Match method is performed so that the final color is similar to the original one in terms of colors. The Least significant bit (LSB) methods along with Spread Spectrum and code based techniques are examples of the spatial techniques [41]. Spatial schemes are simpler and faster than other hiding tech-

niques. However, these methods are less robust, and are easier to be detected by an attacker.

In [19] the authors propose a gray-scale image hiding scheme where the cover image is divided into blocks of two sequential pixels. These blocks are grouped based on their smoothness and contrast attributes. The amount of information that can be hidden in a block is determined by its smoothness and contrast level. Since it is tougher for human perception to detect changes in non-smooth areas, this algorithm suggests to embed more data in edge areas.

Image inpaintaing was also used for data hiding in [26,24]. A scheme that utilizes data-hiding and compression simultaneously using side match vector quantization and image inpainting is proposed in [24]. Since this method combines data hiding and image compression, the maximum capacity reached by this method was quite low (0.14bpp for an R,G,B color secret image). Another class of spatial hiding schemes are the reversible data hiding schemes. Examples of such schemes are the work are [27] and [25].

The second embedding category is the transfer domain methods. In these techniques, an image is transformed from it's spatial domain to a different domain. The hiding process is then performed in the transformed domain by inspecting the less important coefficients, and replace them by scaled bits from the secret image. Finally, a stego image is produced by taking the inverse transform. There are several transforms that are used in image hiding schemes. Examples are, the discrete Fourier transform, discrete cosine transform, and wavelet transform. The advantage of these techniques is the improved security and robustness comparing to the spatial domain techniques.

A frequency domain hiding scheme is introduced in [28,29]. The idea is to use the 'Matryoshka principle' or the 'nested doll principle' which is a well-known design paradigm. The author's theory states that the Fourier phase of the cover image is much more important than its Fourier mgnitude when reconstructing the image. Thus, the scheme keeps the Fourier phase intact and embeds in the Fourier magnitude. This has allowed a robust embedding with acceptable quality of the stego image and minor degradation in the extracted secret image.

The same authors have investigated the trade-off between hiding capacity and perceptibility in many papers. In [32] a Fixed-Block-size locally Adaptive-Region (FBAR) DCT approach was implemented to discover this relationship. An improvement over the previous method using a Fixed-Block-size Globally Adaptive-Region (FB-GAR) DCT method had enabled to embed and extract with higher capacities and perceptibility [33]. Moreover, the authors made a further step in challenging this trade-off in [34]. The new idea is based on partitioning the cover image into non-overlapping segments using a Quad-Tree Adaptive-Region (QTAR) DCT embedding scheme. This has achieved the highest capacity/perceptibility levels among all of their previous work.

A wavelet transform scheme is presented in [36]. This method encrypts the secret data, and then embeds into the wavelet coefficients of the cover image to produce a stego-image. Furthermore, the algorithm increases the robustness

of the proposed method by electing the approximation band of the wavelet domain, and embed in that band. In [11] another novel wavelet approach that embeds hidden information in the integer wavelet coefficients of the cover image. To increase the security, a pseudorandom function is applied to select the coefficients that will be used in hiding.

A third class in hiding schemes is the compression-based algorithms. The author in [35] introduces a methodology that combines Least Significant Bit(LSB), Discrete Cosine Transform(DCT), and compression techniques. At the first place, the secret data is embedded into the cover image using a LSB algorithm to produce a stego image. Then, DCT is used to transform the image to the frequency domain. Finally compression techniques like quantization and runlength coding are performed to compress the stego-image in order to make it more secure.

The embedding scheme introduced in this paper is based on a Curve-Fitting (CF) methodology applied in the transform domain and implemented on the previously published embedding algorithms in [33,34]. It is shown that this simple enhancement will allow for higher embedding capacities while improving the quality of the stego image in comparison to results obtained using these previous schemes. A comparison with the embedding capacity and perceptibility of various steganography schemes that have been recently published in the literature is also demonstrated.

3 Theoretical Background

3.1 The Discrete Cosine Transform

The strong "energy compaction" property has made the Discrete Cosine Transform (DCT) a widely used utility in signal and image processing applications. When using the DCT transform, most of the signal information is concentrated in the top-left area of the domain, which is the area that has low-frequency coefficients[1,37]. Since images are two-dimensional signals, the 2-dimensional DCT (2D-DCT), which is an extension of the 1D-DCT, is often used in image processing.

For images, the strong "energy compaction" property of the DCT manifests clearly with highly correlated images. In contrast, uncorrelated images have their energy spread out in the domain where some large values of the DCT coefficients are found in the high frequency areas.

Another property of the DCT transform is the importance of the DCT-phase component for reconstructing an image in the spatial domain. It has been found that the DCT-phase carries a significant amount of the image information. On the other hand, the DCT-magnitude of the image provides much less significant information about an image [3,4]. Therefore, it is practical to hide in the magnitude of the DCT, and to keep the DCT-phase intact.

The strong energy compaction property of the DCT and the significance of its magnitude and phase spectra can be found with more details in [31] and [33].

3.2 The Discrete Wavelet Transform

The other transform domain that has been widely utilized in recent years in steganography is the discrete wavelet transform (DWT). A Wavelet is a small wave that has its energy concentrated in time to allow dealing with time-varying and non-stationary cases. The Wavelet Transform is a process that can decompose a signal into coefficients that represent the signal within a certain time period [12]. The transform can be defined on an input function $f(t)$ using the following equation:

$$W(R,S) = \int_{-\infty}^{\infty} f(t)\psi_t(R,S)dt, \tag{1}$$

where $W(R,S)$ represents the coefficients which are a function of scale and position transform parameters (R,S), and ψ_t represents the mother wavelet function.

Since images are considered as a discrete two-dimensional signal, the 2D Discrete Wavelet Transform (2D-DWT) must be applied instead of the continuous wavelet transform. The 2D-DWT can be defined on an input image $x(k,j)$ in the form:

$$D(r,s) = \sum_{k}\sum_{j} x(k,j)\psi_{kj}(r,s), \tag{2}$$

where $D(r,s)$ represents the 2D-DWT coefficients as a function of the scale and shift transform parameters (r,s), and ψ_{kj} is the mother wavelet basis time function with finite energy and fast decay.

The DWT decomposition can also be obtained by using filter banks. A filter bank is a series of filters that divides an input signal into multiple components. For images, wavelet analysis can help separate the input image into approximate and detailed sub-images. This can be done by applying a 1D low-pass filter on the rows of the input image, which produces the horizontal approximation of the input image. A 1D high-pass filter is also applied on the rows of the input image, which produces the Horizontal details of the input image. Next, a second round of filters are applied to both the horizontal approximation and the horizontal details, but this time, these filters are applied on the columns. The second filtering stage results in 4 sub-images; the approximation image (LL), the vertical details (LH), the horizontal details (HL), and the diagonal details (HH). Figure 1 illustrates the filter bank operation on an input image.

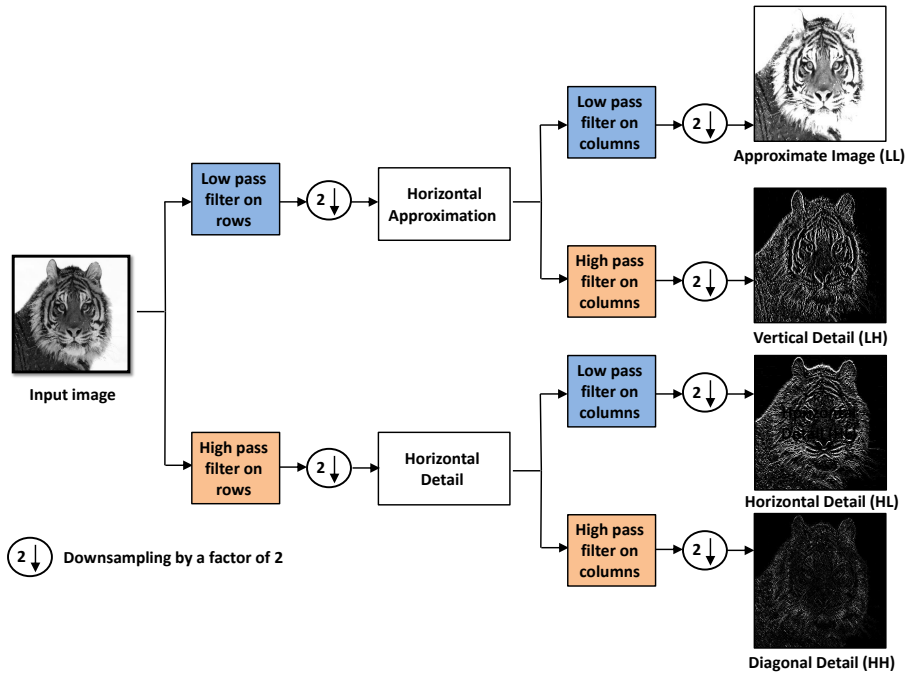

Fig. 1 Steps to obtain the wavelet bands; The Approximate Image (LL), Vertical Detail (LH), Horizontal Detail (HL), and the Diagonal Detail (HH) bands using filter bank.

3.3 Stego Image Quality Measures

Perceptibility is an embedding system aspect that deals with the amount of "distortion" in the cover medium due to the hiding process. The perceptibility level plays an important role in evaluating the performance of an embedding system. It is important in the evaluation to consider both the visual quality of the stego images and the analytical performance of the hiding scheme.

Capturing the existence of an embedded message breaks the fundamental goal of steganography. The definitive measure of visual fidelity are those tests that are related to our human perception. However, these test will give various results since human perception system differs from a person to another.

Tests are performed by people who search for visual differences between the stego and cover images, and trying to detect the original cover image. According to International Telecommunication Union rules and recommendations [13,38], if the percentage of success approaches 50%, then it can be considered that the message is securely hidden.

A well established approach to image fidelity measurement that tries to emulate the human visual perception of image structure, is the Structural SIMilarity (SSIM) index. Under the assumption that human visual perception is highly adapted to extracting structural information from a scene, SSIM was introduced as an alternative complementary framework for quality assessment

based on the degradation of structural information [43,42]. Since the human visual system is more sensitive to changse in the luminance or the contrast channel, this new technique calculates the similarity based on some luminance and contrast measurements.

Contrary to the subjective approach which is based on human perception, another robust measure of image quality that has been widely used by the signal processing community is the Peak-Signal-to-Noise-Ratio (PSNR) in decibels (dB) [7,22]. This measure is less sensitive to minor deviations between images and, together with the SSIM index, will be adopted for measuring the performance of our embedding scheme.

A description of each of these quality measures, including mathematical formulations, can be found with more details in [33,34].

4 The Proposed Curve-Fitting Scheme

The essential feature that attracts researchers to transform domain techniques based on DCT is its energy compaction property. As discussed in section 3, the DCT domain of a correlated image has few coefficients concentrated at the top left corner. These few coefficients represents the low frequency component of an image. As a result, it is guaranteed to have a large area to hide in which lies in the high frequency region of the DCT domain.

The strong compaction property of DCT for highly correlated images is the key feature allowing our new scheme to break the traditional barrier between capacity and stego fidelity. The merits of our scheme is two fold; because our scheme is able to properly utilize the full high frequency areas made available by the DCT's strong energy compaction by utilizing a Curve-Fitting (CF) approach applied in the DCT domain, an optimally large area is available for embedding the secret data, thus allowing for optimally high capacity results. Secondly, since the scheme is utilizing the high frequency areas to embed in, the scheme is able to achieve excellent stego image quality because of the fact that our human visual system is less sensitive to distortion around edges or in high frequency areas.

To clarify this idea, we use the Quad-Tree Adaptive-Region (QTAR) scheme of [34], and the Fixed-Block Global-Adaptive-Region (FB-GAR) method of [33] to show improvements in these methods by re-implementing them using our novel curve-fitting methodology proposed in this work. We thus denote the curve-fitting schemes as (CF-QTAR) and (CF-FB-GAR). To start, CF-QTAR segments the cover image into highly correlated blocks using a quad-tree approach. Then the 2D-DCT is applied to each block to transform the block to the DCT domain. A quantization/thresholding process is performed on each block to obtain a binary image that marks the embedding area.

Unlike previous adaptive region embedding approaches in [32,34,33], where only a block or a segment of the possible hiding area is used to embed in, this scheme fully utilizes the area that is possible to hide in. The CF-QTAR scheme will select 3 points at the edges of the hiding area, and a piecewise linear curve

fits these three point underwhich will lie the entire embedding area. These 3 points are then sent along with the stego image to the receiver side to be able to determine the embedding area, and extract the secret image successfully. This novel curve fitting approach has allowed the embedding algorithms to achieve top capacity results that reached 22bpp at PSNR values that reached 35.83dB.

4.1 The Embedding Process

Segmenting the cover image into coherent regions is the first step in our proposed method. The partitioning technique used follows a quad-tree approach for all of the three (R,G,B) color channels. The quad-tree dividing approach can be classified as a top-down segmentation method that has been widely used in many image processing applications [34, 10].

The quad-tree method starts by dividing an image into four equal sized blocks. Then, it checks the correlation level in each pixel. This check is done by simply computing the difference between the maximum pixel value and the minimum pixel value in a block. If the difference value is greater than a pre-defined "threshold" value, then this block is further sub-divided into four blocks. Otherwise, the block is assumed that it has met the pre-defined coherence standard, and it is not divided any more. This process is performed repeatedly until each block meets the criterion.

Along with the threshold value, there are two more quad-tree parameters; the maximum block size, and the minimum block size. The quad-tree algorithm will not sub-divide a block to a lower block size than the minimum block size even if the block did not meet the criterion. In the same way, the algorithm will force segmenting a block that is larger than the maximum block size even if the difference between the maximum and minimum pixel value in that block is less than the threshold value.

After partitioning the cover image using a quad-tree approach, the 2D-DCT is applied to each block. Next, a quantization process begins to estimate the embedding region in each block. The embedding process is performed by replacing the unimportant DCT coefficients by a scaled version of the secret image. The inverse DCT is then applied to each pixel producing the stego image in the spatial domain. Figure 2 shows a general demonstration of the embedding process.

The quantization process is used to locate the region that is appropriate for embedding. CF-QTAR uses a quantization technique that is similar to the one used in JPEG compression standard. The idea is to choose one of the JPEG quantization matrices, and divide the magnitude of the DCT of the current block with the quantization matrix values element by element, after first resizing the quantization matrix to the same size as the current block. We use the Matlab function "imresize" to resize the original 8×8 JPEG quatization matrix, shown in step-1 of figure 3, to the same size as the current block size of the quad-tree-segmented cover image. The main purpose of this quantization

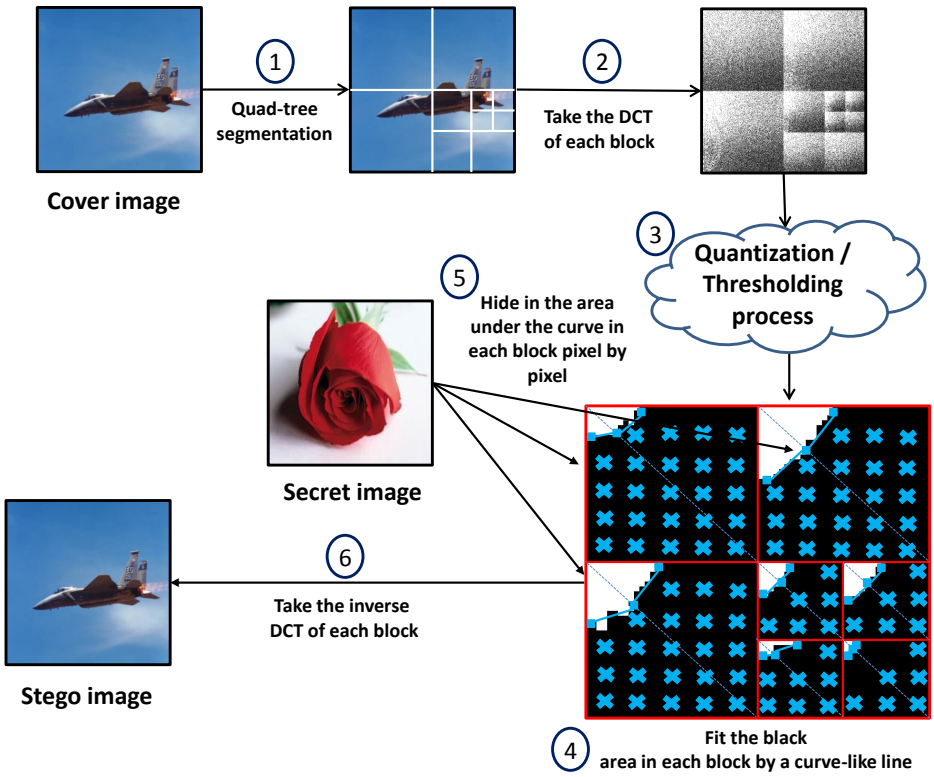

Fig. 2 A general illustration of the hiding procedure.

step is to distinguish between important and redundant DCT coefficients, masking out the less important coefficients and replace them with the scaled secret information while keeping the important DCT coefficients intact, so that the scheme can reconstruct the cover image with minimal distortion, which is of utmost improtance for the security of steganography applications.

After quantizing the magnitude of the DCT of a block, a thresholding process begins. The quantized DCT block is divided into 8×8 non-overlapping segments, and the values inside each segment is checked. If a non-zero value is found in the segment, then the whole segment is turned to white. Otherwise, the block is kept unchanged if all values inside this block are zeros. This process converts the quantized matrix to a binary image, where the black area (zeros) is the proposed embedding area. On the other hand, the white region indicates the area that contains the important DCT coefficients that must be kept intact. The quantization and thresholding step is shown in figure 3.

The pre-embedding step is to fit the black area by a piecewise linear curve. This is done by locating 3 points; points 1 and 3 are located at the edges of the black area, point 2 is the intersection between the diagonal and the black area. These points are then connected by a line generated by a linear

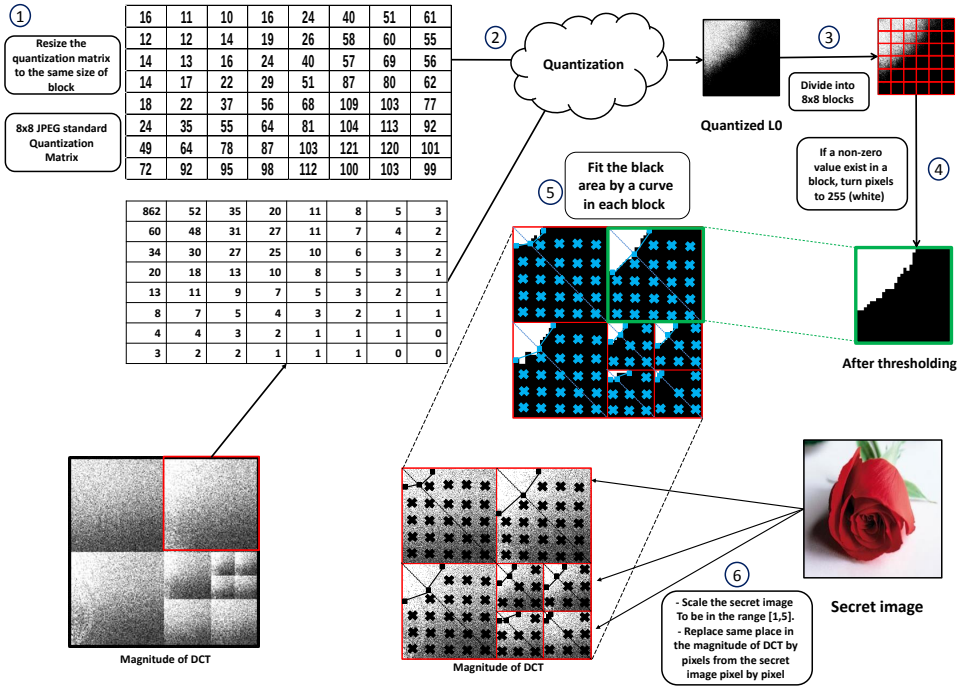

Fig. 3 The quantization and thresholding step.

interpolation algorithm. The black area must be now bordered by the line from top, and thus the region under this line is the embedding area. Figure 4 shows the curve-fitting procedure. These points must be sent to the receiver so that it can locate the embedding area, and extract the secret image. The maximum square size of the secret image that can fit into the proposed area can be estimated as follows:

$$\mathrm{S\ x\ S} = \left\lfloor \sqrt{\sum_{m} b_m} \right\rfloor , \tag{3}$$

where S is the maximum dimension of the square secret image, m is the index of quad-tree blocks, b_m is the number of pixels that lie under the piecewise linear curve in block $b_m; m = 1, 2, 3, \cdots, N$ blocks.

Finally, the embedding is done in the magnitude of the DCT in the locations of the black areas of the quantized matrix after thresholding. Coefficients are replaced by scaled pixels from the secret image in the range [1,5] pixel by pixel column-wise from the top left to the bottom right. The re-scaling step is important to allow the hidden secret image values to blend into the natural range of values of the DCT coefficients. Figure 5 shows the pseudocode for the embedding process.

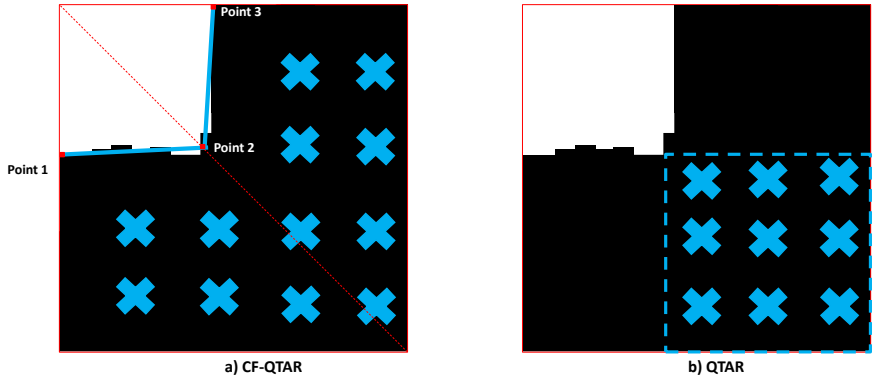

Fig. 4 (a) The black region represents the area that has less important DCT coefficients in a Quad-Tree block, and can be replaced by scaled pixel values from the secret image without sacrificing the stego image quality. (b) In QTAR [34], the embedding area is selected by finding the largest square block that can fit in the lower-left corner of the black area. On the other hand, CF-QTAR utilizes the whole proposed area by fitting a piecewise linear curve to three points bordering the region as shown.

4.2 The DWT-QTAR and CF-DWT-QTAR Schemes

In this section, we investigate our Curve-Fitting hiding approach using wavelets. DWT-QTAR is a modified version of QTAR where the DWT of the quad-tree blocks is used instead of the DCT. First, the cover image is segmented in a quad-tree segmentation fashion similar to QTAR. Then DWT is applied on each block. This operation produces 4 sub blocks in each quad-tree block, namely the Approximate Image (LL), Vertical Detail (LH), Horizontal Detail (HL), and Diagonal Detail (HH). After that, the LL band is transformed to the frequency domain using DCT. This will make the whole quad-tree block represented in the frequency domain. The magnitude of the DCT of the LL band is then quantized to find the area (P) that has the least important DCT coefficients. Then, a contiguous square region in the lower-right corner of this (P) area is selected to be a feasible area to hide in. This square region is embedded with part if the secret image and the rest of the secret image pixels will replace the DWT coefficients of the LH, HL, and HH bands. Figure 6 clarifies the idea of DWT-QTAR.

CF-DWT-QTAR on the other hand, is an implementation of our Curve-Fitting approach on the DWT-QTAR technique. CF-DWT-QTAR tries to utilize the whole feasible area (P) in the LL band, instead of hiding in a square region which would not fully utilize the whole area (P) for hiding. Figure 7 presents the idea of CF-DWT-QTAR. The pseudocode of the CF-DWT-QTAR embedding process is presented in figure 8.

```
Algorithm: [f'] = CFQTAR (im, sec)
Input:     im:  cover host image
           sec: secret message image to be hidden
           QM: 8x8 quantization matrix values
           n: minimum quad-tree block size [n x n]
           m: maximum quad-tree block size [m x m]
           s: scale factor empirically choosen between [1,5]
           t: threshold value
Output:    f': stego image containing the secret image
 1: [imquad, N] ← Quad(im,m,n,t) //quad-tree segmentation into N blocks
 2: for i := 1 to N quad-tree blocks
 3:   DCT_MAG ← abs(dct2(imquad)) // apply 2D DCT on each block, and get the magnitude
 4:   DCT_PHASE ← angle(dct2(imquad)) // apply 2D DCT on each block, and get the phase
 5:   resize(DCT_MAG,QM) // resize QM to the same size of DCT_MAG
 6:   Qi ← DCT_MAG/QM // quantize each block by dividing by QM
 7:   Qu' ← divide(Qi, 8) //divide the quanitized matrix into K [8 x 8] blocks
 8:   for i := 1 to K blocks
 9:      non_zeros ← size(Qi' ~= 0) // get the number of non-zeros in each block
10:      if non_zeros > 0
11:         Qu' = 255 // set the whole block to 255 (white)
12:      end if
13:      Qu ← pack(Qu')
14:   end for
15: end for
16: index ← find(Qu == 0) // locate the embedding areas in each block
17: no_zero ← size(index) // find the total number of zeros in Qu
18: size_sec  ← floor(sqrt(no_zero)) // get the maximum secret size
19: sec' ← resize(sec, size_sec) // resize the secret image to [size_sec x size_sec]
20: sec_sc ← rescale(sec', [1,s]) // rescale the secret image to be in the range [1,s]
21: for i := 1 to [size_sec x size_sec]
22:    DCT_MAG'← replace(DCT_MAG,sec_sc,index) // replace DCT_MAG with sec_sc at index
23:    IMGDCT ← repack(DCT_MAG',DCT_PHASE) // recreate the full DCT
24:    fi ← idct2(IMGDCT) // apply inverse 2D DCT on IMGDCT in each block
25: end for
26: f ← pack(fi)
27: return f'  // the final stego image
```

Fig. 5 The pseudocode of the CF-QTAR embedding process.

4.3 The CF-FB-GAR Scheme

A special case of the QTAR scheme of [34] is the FB-GAR scheme introduced in [33], which differs from QTAR by setting the block size for the maximum and minimum quad-tree blocks to the same value. This segments the image into fixed-size blocks. Similar improvements as those achieved by the CF-QTAR can also be obtained for the FB-GAR scheme by using our curve-fitting methodology. We call this the curve-fitted fixed-block global-adaptive-region (CF-FB-GAR) scheme.

Wavelets can also be applied to the FB-GAR scheme of [33] using the same approach described in section 4.2 for QTAR. We call this DWT-FB-GAR, and experiemental results in section 5 will show that our curve-fitting technique when applied to these adaptive-region embedding schemes will improve the visual quality of the stego image while allowing higher capacities for data hiding.

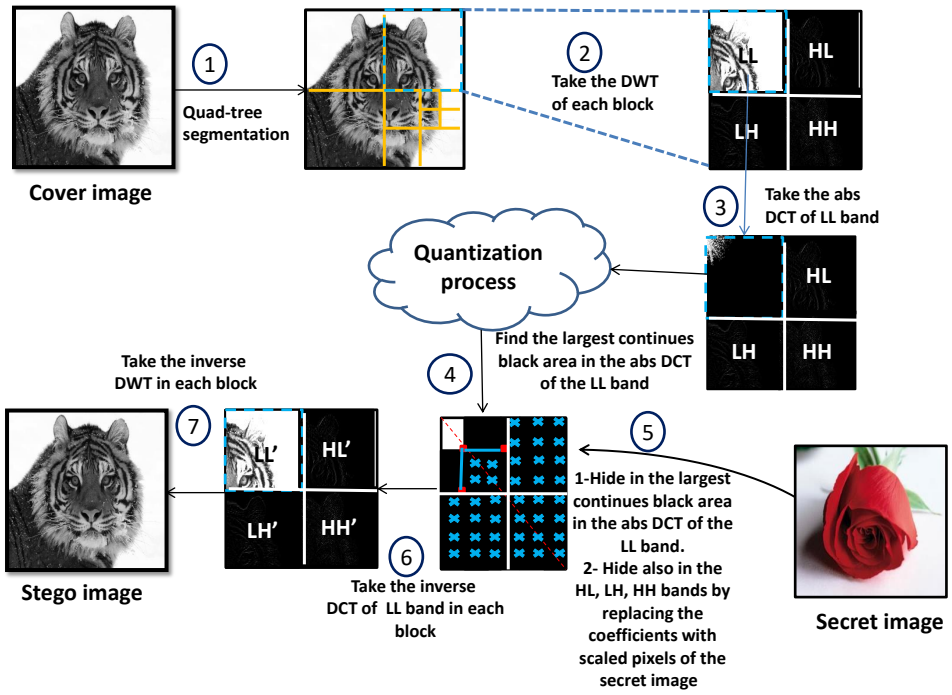

Fig. 6 An illustration of the DWT-QTAR approach.

4.4 The CF-QTAR Extraction Process

To successfully extract the secret image, the size of the adaptive blocks, their locations, and their 3 points must be transmitted along with the stego image. Extraction is performed in the reverse order as follows:

- The received stego image is segmented into quad-tree blocks using the received size and locations of each block.
- The 2D-DCT of each block is computed and the magnitude of this DCT is obtained.
- A piece-wise linear curve is fitted to each block's 3 points (which are received with the stego image), as shown in figure 4.
- The region under each block's curve is assumed to have the secret data. The secret data is extracted from each block pixel by pixel column-wise from the top left to the bottom right, in the same order that was used during the embedding process.
- Finally, the values of the pixels must be rescaled from the current range to the original intensity range of $[0, 255]$ per color channel.

Figures 9 and 10 show the pseudocode of the extraction process for both CF-QTAR and CF-DWT-QTAR methods.

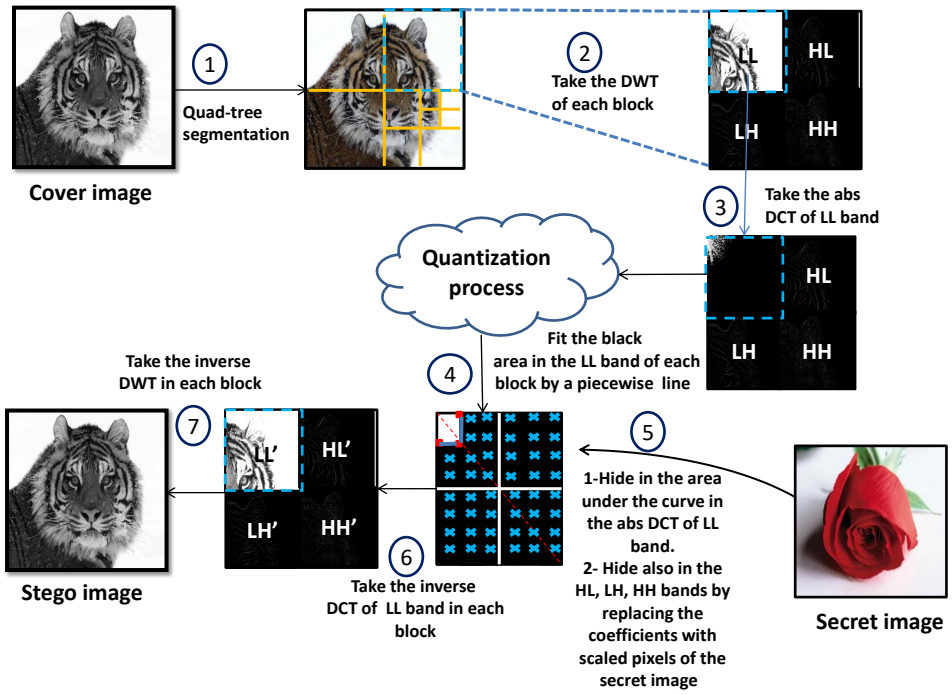

Fig. 7 An illustration of the CF-DWT-QTAR approach.

5 Experimental Results

In this part, we compare results obtained by utilizing our curve-fitting embedding scheme against various other steganography schemes which have been recently published in the literature. Detailed demonstrative results of the CF-FB-GAR, CF-QTAR, and CF-DWT schemes applied at different block-size dimensions of $\{32 \times 32, 64 \times 64, 128 \times 128\}$, and tested on four different color cover host images are presented. Figure 11 shows the four different cover images used: "Balloons", "TigerPounce", "F15Large", and "Zebras". The three secret images used, "Flower", "Handwriting", and "Pasta" are also shown in the same figure.

5.1 Comparative Results

For the fixed-block adaptive-region (FBAR) embedding scheme proposed in [32], the authors were able to embed, in a typical cover image of a natural scene, and losslessly extract, from the generated stego image, approximately 6.74bpp per color channel of a three channel (R,G,B) color cover image for a maximum overall 20.22bpp embedding capacity with a perceptibility measured at PSNR of 25dB. This corresponds to embedding a color image of size 470×470 inside a color cover image of size 512×512.

```
Algorithm: [f'] = CFDWTQTAR (im, sec)
Input:     im:  cover host image
           sec: secret message image to be hidden
           QM: 8x8 quantization matrix values
           n: minimum quad-tree block size [n x n]
           m: maximum quad-tree block size [m x m]
           s: scale factor empirically choosen between [1,5]
           t: threshold value
           wavelet: The mother wavelet to be used
Output:    f': stego image containing the secret image
 1: [imquad, N] ← Quad(im,m,n,t) //quad-tree segmentation into N blocks
 2: for i := 1 to N quad-tree blocks
 3:   [LL LH HL HH] ← dwt2(imquad,wavelet) // apply DWT on each quad-tree block
 4:   DCT_MAG ← abs(dct2(LL)) // apply 2D DCT on LL band, and get the magnitude
 5:   DCT_PHASE ← angle(dct2(imquad)) // apply 2D DCT on LL band, and get the phase
 6:   resize(DCT_MAG,QM) // resize QM to the same size of DCT_MAG
 7:   Qi ← DCT_MAG/QM // quantize each block by dividing by QM
 8:   Qu' ← divide(Qi, 8) //divide the quanitized matrix into K [8 x 8] blocks
 9:   for i := 1 to K blocks
10:     non_zeros ← size(Qi' ~= 0) // get the number of non-zeros in each block
11:     if non_zeros > 0
12:       Qu' = 255 // set the whole block to 255 (white)
13:     end if
14:     Qu ← pack(Qu')
15:   end for
16: end for
17: index ← find(Qu == 0) // locate the embedding areas in each block
18: no_zero ← size(index) // find the total number of zeros in Qu
19: size_sec_LL  ← floor(sqrt(no_zero)) // get the maximum secret size to hide in LL band
20: size_sec  ← size_sec_LL + size(LH) + size(HL) + size (HH) // get the maximum secret size
21: sec' ← resize(sec, size_sec) // resize the secret image to [size_sec x size_sec]
22: sec_sc ← rescale(sec', [1,s]) // rescale the secret image to be in the range [1,s]
23: for i := 1 to [size_sec_LL x size_sec_LL]
24:   DCT_MAG'← replace(DCT_MAG,sec_sc,index) // replace DCT_MAG with pixels from sec_sc at index
25:   IMGDCT ← repack(DCT_MAG',DCT_PHASE) // recreate the full DCT
26:   fi ← idct2(IMGDCT) // apply inverse 2D DCT on IMGDCT in each block
27: end for
28: [LH' HL' HH'] ← replace([LH HL HH],sec_sc)  // replace LH,HL, and HH bands with pixels from sec_sc
29: stego_block  ← idwt2([fi LH' HL' HH', wavelet) // Take the inverse dwt of each block
30: f ← pack(stego_block)
31: return f'  // the final stego image
```

Fig. 8 The pseudocode of CF-DWT-QTAR embedding process.

```
Algorithm: [sec] = CFQTAR_extract (stego,index_blocks,sizes_blocks,CF_POINTS)
Input:     stego:  Stego image
           index_blocks: Indices of the quad-tree blocks
           sizes_blocks: Sizes of quad-tree blocks
           CF_POINTS: 3 points used to construct a curve. The area under this curve contains the secret image
Output:    sec: The extracted sercret image
1: [imquad, N] ← Quad(stego,index_blocks,sizes_blocks) //quad-tree segmentation into N blocks
2: for i := 1 to N quad-tree blocks
3:   DCT_MAG ← abs(dct2(imquad)) // apply 2D DCT on each block, and get the magnitude
4:   imsec_block ← extract(DCT_MAG,CF_POINTS) // construct a curve using CF_POINTS, and retrieve pixels under the curve
5: end
6: imsec ← pack(imsec_block) // get the secret image
7: sec ← rescale(imsec, 255) // rescale back to be in range [0 255]
```

Fig. 9 The pseudocode of CF-QTAR extraction process.

The fixed-block global-adaptive-region (FB-GAR) scheme, described in [33], was able to embed at a capacity of 20.83bpp with a fixed block size of 128×128 and for the "F15Large" cover image at a PSNR of 27.24 dB.

The quad-tree-adaptive-region (QTAR) embedding scheme, proposed in [34], was able to introduce improvements over the FB-GAR scheme for the same perceptibility PSNR of the stego image. It is able to embed, in a typical cover image of a natural scene, and losslessly extract, from the generated stego image, approximately 7bpp per color channel of a three channel (R,G,B) color cover image for a maximum overall 21.01bpp embedding capacity for the same cover image (the "F15Large" cover image) when setting the minimum quad-

```
Algorithm: [sec] = CFDWTQTAR_extract (stego,index_blocks,sizes_blocks,CF_POINTS)
Input:     stego:  Stego image
           index_blocks: Indices of the quad-tree blocks
           sizes_blocks: Sizes of quad-tree blocks
           CF_POINTS: 3 points used to construct a curve. The area under this curve contains the secret image
Output:    sec: The extracted secret image
 1: [imquad, N] ← Quad(stego,index_blocks,sizes_blocks) //quad-tree segmentation into N blocks
 2: for i := 1 to N quad-tree blocks
 3:   [LL LH HL HH] ← dwt2(imquad,wavelet) // apply DWT on each quad-tree block
 4:   DCT_MAG ← abs(dct2(LL)) // apply 2D DCT on LL band, and get the magnitude
 5:   LL_sec ← extract(DCT_MAG,CF_POINTS) // construct a curve using CF_POINTS, and retrieve pixels under the curve
 6:   imsec_block ← pack([LL_sec LH HL HH]) // get secret pixels embedded in this block
 7: end
 8: imsec ← pack(imsec_block) // get the secret image
 9: sec ← rescale(imsec, 255) // rescale back to be in range [0 255]
```

Fig. 10 The pseudocode of CF-DWT-QTAR extraction process.

Fig. 11 (a-d) The four 512 x 512 cover image used, from left to right: "Balloons", "Tiger-Pounce", "F15large", and "Zebras". (e-g) The three secret images: "Flower", "Handwriting", and "Pasta" are shown respectively.

tree block size to 128×128 and maximum to 256×256, with a perceptibility measured at PSNR of 27.21 dB. This corresponds to embedding a secret color image of size 479×479 inside a color cover image of size 512×512.

Table 1 shows the different embedding capacities and PSNR values reached by some recent state-of-the-art schemes in comparison to our curve-fitting (CF) methodology implemented in both the QTAR scheme (CF-QTAR and CF-DWT-QTAR) and the FB-GAR scheme (CF-FB-GAR). The table clearly shows that the high capacity results achieved by our scheme is the highest among all of our previous work. The scheme is able to embed approximately 22.7bpp for the "F15Large" cover image. This is equivalent to hide a 498 x 498 color image into a 512×512 color cover image. The quality of the cover image is measured to be 28.15 db, which is even higher than other lower capacity schemes. Another promising result obtained by our curve-fitting CF-QTAR scheme is a capacity rate of 19.88bpp at a PSNR of 35.02 db for the "Balloons" cover image and a scale size of 32×32, which is an improvement over the capacity of the curve-fitting CF-FB-GAR scheme for the same "Balloons" cover image and at the same scale size of 32×32 achieving only 19.54bpp for the same visual stego quality of 35.03 db, as is clear from table 1.

5.2 Analysis of Results

A sample of the results demonstrating our CF-QTAR curve-fitting scheme are shown in figures 12-17. The "Flower" image was the embedded secret image used in these example results. Tables 2 to 9 illustrate comparative results between our proposed curve-fitting approach for both DCT and DWT transform domains (namely; CF-FB-GAR, CF-DWT-FB-GAR, CF-QTAR, CF-DWT-QTAR) with the DCT-based Fixed-Block Global-Adaptive-Region (FB-GAR)

Table 1 Comparative results expressed as maximum Capacity/PSNR values for the various methods. Highest Capacities and PSNR values are emphasized in a bold font.

Method	Capacity	PSNR
Lee & Chen (2000) [15]	**12.18 bpp**	34.03 dB
Yang *et. al.* (2004) [44]	1.96 bpp	28.16 dB
Brisbane *et. al.* (2005) [5]	6 bpp	**40 dB**
Lin & Shiu (2010) [18]	1.02 bpp	28.22 dB
Rabie (FFT) (2013) [30]	6 bpp	19.51 dB
Qin *et. al.* (2015) [25]	3.48 bpp	**41 dB**
Rabie & Kamel FBAR (2015) [32]	20.22 bpp	25 dB
Rabie & Kamel FB-GAR (2016) [33]	**20.83 bpp**	27 dB
Rabie & Kamel QTAR (32 x 32) [34]	**19.29 bpp**	**27.89 dB**
Rabie & Kamel QTAR (64 x 64) [34]	**19.97 bpp**	**27.73 dB**
Rabie & Kamel QTAR (128 x 128) [34]	**21.01 bpp**	27 dB
CF-FB-GAR (32 x 32)	**19.54 bpp**	**35.03 dB**
CF-FB-GAR (64 x 64)	**22.43 bpp**	28.49 dB
CF-FB-GAR (128 x 128)	**20.83 bpp**	**32.54 dB**
CF-QTAR (32 x 32) (Max. PSNR)	**19.88 bpp**	**35.02 dB**
CF-QTAR (32 x 32) (Max. Capacity)	**22.70 bpp**	28.15 dB
CF-QTAR (64 x 64) (Max. PSNR)	**20.05 bpp**	**34.37 dB**
CF-QTAR (64 x 64) (Max. Capacity)	**22.61 bpp**	28.23 dB
CF-QTAR (128 x 128) (Max. PSNR)	**19.79 bpp**	**34.0 dB**
CF-QTAR (128 x 128) (Max. Capacity)	**22.52 bpp**	28.4 dB
CF-DWT-QTAR (32 x 32) (Max. PSNR)	**21.26 bpp**	**37.85 dB**
CF-DWT-QTAR (32 x 32) (Max. Capacity)	**22.52 bpp**	33.80 dB
CF-DWT-QTAR (64 x 64) (Max. PSNR)	**22.0 bpp**	**35.83 dB**
CF-DWT-QTAR (64 x 64) (Max. Capacity)	**22.0 bpp**	**35.83 dB**
CF-DWT-QTAR (128 x 128) (Max. PSNR)	18.77 bpp	**43.41 dB**
CF-DWT-QTAR (128 x 128) (Max. Capacity)	**20.5 bpp**	31.06 dB

scheme of [33], the DCT-based Quad-Tree-Adaptive-Region (QTAR) scheme of [34], the DWT-FB-GAR and DWT-QTAR approaches implemented in section 3.2. For the quad-tree approaches, experiments were done at various minimum quad-tree block sizes, but were allowed to grow to a maximum of 256×256.

The cover images used "Balloons", "TigerPounce", "F15Large", and "Zebras" have different levels of correlation and are all of size 512×512. Since "F15Large" has the largest highly correlated area of all the cover images used, the capacity achieved using this image as a cover image was the highest among all the other cover images. The capacity reached an upper limit of 22.7bpp which is equivalent to embedding a secret image of size 498×498. When using the "Zebras" as a cover image, it is clear from the tables that this cover image gave the lowest capacity (19.22bpp) due to the highly uncorrelated nature of the image (too much high-frequency regions). This is a natural consequence

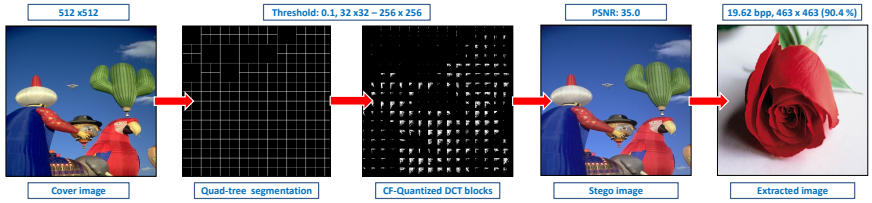

Fig. 12 Hiding the secret image "Flower" into 512 x 512 "Balloons" cover image. From left to right: 1) The cover image, 2) Quad-tree segmentation with threshold value of 0.1, 3) The curve-fitted DCT of each block after quantization 4) The stego image with 35.0 dB, 5) The extracted secret image with size of 463 x 463 (19.62bpp)

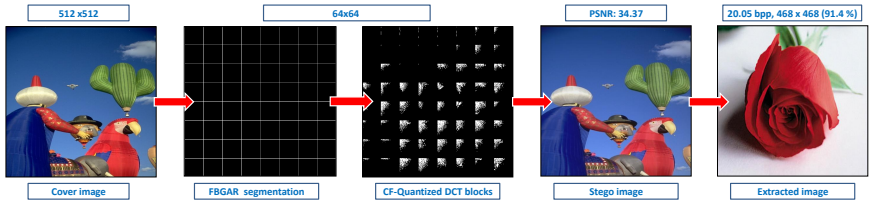

Fig. 13 Hiding the secret image "Flower" into 512 x 512 "Balloons" cover image using CF-FB-GAR. From left to right: 1) The cover image, 2) FB-GAR segmentation using 64×64 block size, 3) The curve-fitted DCT of each block after quantization 4) The stego image with 34.37 dB, 5) The extracted secret image with size of 468 x 468 (20.05bpp)

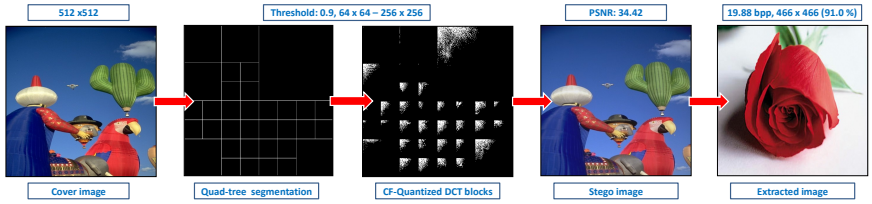

Fig. 14 Hiding the secret image "Flower" into 512 x 512 "Balloons" cover image. From left to right: 1) The cover image, 2) Quad-tree segmentation with threshold value of 0.9, 3) The curve-fitted DCT of each block after quantization, 4) The stego image with 34.42 dB, 5) The extracted secret image with size of 466 x 466 (19.88bpp).

of the type of quad-tree segmentation used and shows that our technique is more suitable to highly correlated cover images, which is a reasonable choice if high capacity is desired.

The experimental results and comparisons clarify that our curve-fitting scheme has exceeded the maximum capacity level achieved by our earlier techniques without sacrificing the stego image quality. Analysis of the data from tables 2 - 9 shows that *91.97%* of the results of the curve-fitting technique have improved capacities without sacrificing the quality of the stego images. To be more specific, *73.24%* of this *91.97%* have also improved the quality of the stego image, which represents *67.36%* of the total recorded results. Our statistics shows that only *4.47%* of the results have higher capacities at lower PSNR values. Only *3.56%* of the results did not show any improvement neither in the capacity nor in the quality of the stego image.

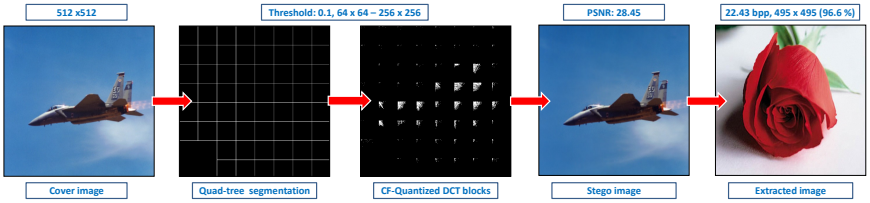

Fig. 15 Hiding the secret image "Flower" into 512 x 512 "F15Large" cover image. From left to right: 1) The cover image, 2) Quad-tree segmentation with threshold value of 0.1, 3) The curve-fitted DCT of each block after quantization, 4) The stego image with 28.45 dB, 5) The extracted secret image with size of 495 x 495 (22.43bpp).

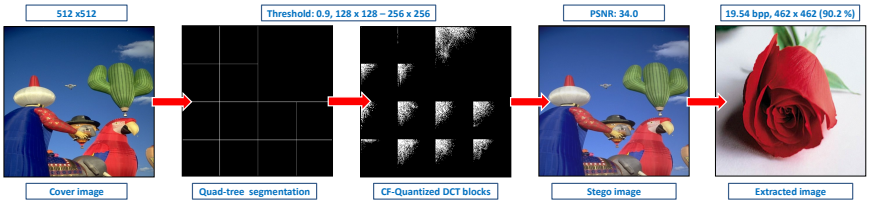

Fig. 16 Hiding the secret image "Flower" into 512 x 512 "Balloons" cover image. From left to right: 1) The cover image, 2) Quad-tree segmentation with threshold value of 0.9, 3) The curve-fitted DCT of each block after quantization, 4) The stego image with 34.0 dB, 5) The extracted secret image with size of 462 x 462 (19.54bpp).

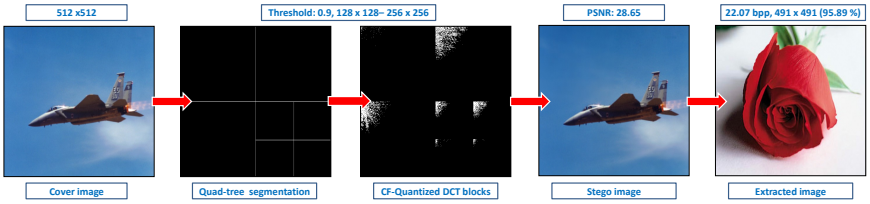

Fig. 17 Hiding the secret image "Flower" into 512 x 512 "F15Large" cover image. From left to right: 1) The cover image, 2) Quad-tree segmentation with threshold value of 0.9, 3) The curve-fitted DCT of each block after quantization, 4) The stego image with 28.65bpp, 5) The extracted secret image with size of 491 x 491 (22.07bpp).

However, the DWT methods (DWT-FB-GAR versus CF-DWT-FB-GAR, and DWT-QTAR versus CF-DWT-QTAR) did not show marked improvements while comparing between the CF and non-CF versions of these DWT methods. The reason for this can be understood by referring to the embedding procedure of both methods, which are clarified in figures 6, and 7. Both methods are hiding in the HH, HL, and LH bands in each block. The only difference is in the LL band, which is 1/4 of a quad-tree block in size. This means that the difference in the embedding area will not be too significant.

Another important note is that the curve-fitting approach has shown its best improvement when applying it to QTAR. For example, referring to tables 6, and 7, when using the "TigerPounce" as a cover image, the "Flower" as a secret image, and when setting the minimum block size to 32×32, we have reached a PSNR value of 28.85dB with a capacity rate of 18.8bpp in QTAR. In

CF-QTAR, although the capacity has increased to 20.74bpp, the PSNR was also raised to 32.88dB.

Another noticeable result is when using 128×128 as the minimum block size with the "Balloons" cover image. The quality of the stego image has improved from 31.81 dB to 34.23 dB with also an increase in the capacity rate from 18.29bpp to 19.54bpp. CF-FB-GAR method comes in the second rank after CF-QTAR. Referring to tables 2, and 3, the PSNR has grew from 27.23 dB to 32.54 dB while an increase also in the capacity rate from 19.63bpp to 20.83bpp. This result was obtained by setting the minimum block size to 128×128, using the "TigerPounce" as the cover image, and "Flower" as the secret image.

Table 10 illustrates detailed results obtained by the CF-QTAR scheme at various threshold values not used in table 7 which produce improved results. The highest capacity rate reached by CF-QTAR was 22.70bpp with a PSNR value of 28.15 dB. This is achieved when setting the minimum block size to 32×32 and using the 'F15Large" as the cover image. On the other hand, CF-QTAR was able to embed the "Flower" secret image into the "Balloons" cover image with a stego image PSNR of 35.02 dB and a capacity rate of 19.88bpp. This result has the highest PSNR value, and is obtained by setting the minimum block size to 32×32. Another attractive result obtained by CF-QTAR is when using the "Balloons" cover image with a minimum block size of 64×64. The capacity rate reached up to 20.05bpp while having a stego image PSNR value of 34.37 dB.

5.3 The Curve-Fitting Advantage

Although CF-QTAR/QTAR schemes and CF-FB-GAR/FB-GAR schemes share the same initial segmentation procedure, the curve-fitting implementations improved the embedding capacity while enhancing the perceptual quality level. The main reason for this improvement is clarified in figure 4. Curve-fitting allows for utilizing the whole proposed embedding area, clearly shown in figure 4-(a). As such, the capacity will increase naturally.

On the other hand, the reason that curve-fitting schemes achieve higher PSNR is the uniform embedding of the scaled secret image values that allow it to blend into the natural range of values of DCT coefficients that it replaces. This embedding takes place in the *whole area under the curve* for curve-fitting schemes, as compared to embedding in just a square region in the lower-right corner of the DCT for non-curve-fitting schemes (for example compare figure 7 and figure 6), which causes abrupt changes in DCT coefficient values between the square region used to embed the scaled secret data and the rest of the DCT coefficients, as is clear from figure 4-(b). When transforming back the DCT coefficients to the space domain, the uniform embedding for curve-fitting schemes will cause less noise artifacts to be generated in the stego image, thus contributing to its improved visual quality.

Table 2 This table presents results from **FB-GAR**. Highest PSNR and Capacity values are emphasized in a bold font.

Block size	Cover	Secret	PSNR	SSIM	Capacity
32 x 32	*Balloons*	*Flower*	**33.37**	0.9708	16.23
		Handwriting	32.5	0.9579	16.22
		Pasta	32.54	0.9579	16.22
	TigerPounce	*Flower*	28.71	**0.9994**	17.17
		Handwriting	33.30	0.9579	17.16
		Pasta	31.15	0.9991	17.16
	F15Large	*Flower*	27.95	0.9871	18.05
		Handwriting	28.18	0.9767	18.04
		Pasta	27.07	0.9830	18.04
	Zebras	*Flower*	23.31	0.9467	15.24
		Handwriting	22.61	0.9298	15.24
		Pasta	23.06	0.9320	15.24
64 x 64	*Balloons*	*Flower*	32.51	0.9844	17.25
		Handwriting	31.4	0.9579	17.24
		Pasta	31.54	0.9872	17.24
	TigerPounce	*Flower*	27.64	0.9968	19.04
		Handwriting	30.67	0.9579	19.04
		Pasta	29.34	0.9961	19.03
	F15Large	*Flower*	27.67	0.9851	19.29
		Handwriting	27.35	0.9787	19.29
		Pasta	26.46	0.9815	19.29
	Zebras	*Flower*	19.74	0.9553	15.24
		Handwriting	19.27	0.9518	15.24
		Pasta	19.56	0.9536	15.24
128 x 128	*Balloons*	*Flower*	31.15	0.9212	18.21
		Handwriting	29.37	0.9237	18.54
		Pasta	29.56	0.9352	18.54
	TigerPounce	*Flower*	27.23	0.9967	19.63
		Handwriting	29.97	0.9963	19.62
		Pasta	28.80	0.9579	19.63
	F15Large	*Flower*	27.24	0.9766	**20.83**
		Handwriting	30.67	0.9579	19.04
		Pasta	25.62	0.9784	20.83
	Zebras	*Flower*	16.53	0.8437	15.24
		Handwriting	16.23	0.8453	15.24
		Pasta	16.43	0.8499	15.24

Table 3 This table presents results from **CF-FB-GAR**. Highest PSNR and Capacity values are emphasized in a bold font.

Block size	Cover	Secret	PSNR	SSIM	Capacity
32 x 32	*Balloons*	*Flower*	**35.03**	0.9787	19.54
		Handwriting	34.57	0.9675	19.54
		Pasta	34.91	0.9699	19.54
	TigerPounce	*Flower*	33.39	0.9996	19.96
		Handwriting	34.89	0.9995	19.97
		Pasta	34.85	0.9995	19.96
	F15Large	*Flower*	29.20	0.9875	**21.71**
		Handwriting	28.56	0.9859	21.71
		Pasta	29.67	0.9879	21.71
	Zebras	*Flower*	28.5	0.9652	15.24
		Handwriting	27.75	0.9629	15.5
		Pasta	28.21	0.9629	15.4
64 x 64	*Balloons*	*Flower*	27.95	0.9871	18.05
		Handwriting	33.73	0.9689	20.05
		Pasta	34.23	0.9661	20.05
	TigerPounce	*Flower*	32.73	0.9994	20.74
		Handwriting	33.50	0.9991	20.74
		Pasta	33.71	0.9992	20.74
	F15Large	*Flower*	28.49	0.9896	22.43
		Handwriting	27.63	0.9848	22.43
		Pasta	28.73	0.9871	22.43
	Zebras	*Flower*	27.47	0.9644	15.84
		Handwriting	26.75	0.9629	15.84
		Pasta	27.18	0.9645	15.84
128 x 128	*Balloons*	*Flower*	34.0	0.9767	19.79
		Handwriting	33.29	0.9623	19.79
		Pasta	28.73	0.9621	19.79
	TigerPounce	*Flower*	32.54	0.9995	20.83
		Handwriting	33.22	0.9992	20.83
		Pasta	33.44	0.9994	20.83
	F15Large	*Flower*	28.15	0.9812	**22.52**
		Handwriting	27.35	0.9867	22.52
		Pasta	28.37	0.9894	22.52
	Zebras	*Flower*	27.45	0.9743	15.76
		Handwriting	26.72	0.9615	15.77
		Pasta	27.16	0.9629	15.76

Table 4 This table presents results from **DWT-FB-GAR**. Highest PSNR and Capacity values are emphasized in a bold font.

Block size	Cover	Secret	PSNR	SSIM	Capacity	Wavelet
32 x 32	*Balloons*	*Flower*	32.17	0.9840	18.1	bior1.1
		Handwriting	32.61	0.9843	18.1	
		Pasta	32.35	0.9842	18.1	
	TigerPounce	*Flower*	32.34	0.9993	**19.33**	coif5
		Handwriting	32.31	0.9995	18.83	
		Pasta	32.01	0.9995	18.83	
	F15Large	*Flower*	**42.06**	0.9895	18.11	db7
		Handwriting	40.5	0.9954	18.1	
		Pasta	42.3	0.9922	18.13	
	Zebras	*Flower*	23.12	0.9786	18.1	db7
		Handwriting	23.16	0.9787	18.1	
		Pasta	23.0	0.9786	18.1	
64 x 64	*Balloons*	*Flower*	32.16	0.9838	18.02	bior1.1
		Handwriting	32.61	0.9843	18.1	
		Pasta	32.35	0.9842	18.1	
	TigerPounce	*Flower*	33.60	0.9995	18.02	bior4.4
		Handwriting	33.1	0.9994	19.15	
		Pasta	33.1	0.9994	19.15	
	F15Large	*Flower*	**41.62**	0.9922	18.02	bior4.4
		Handwriting	41.2	0.9953	18.4	
		Pasta	40.58	0.9931	18.41	
	Zebras	*Flower*	22.57	0.9778	19.03	coif5
		Handwriting	22.86	0.9778	19.04	
		Pasta	22.6	0.9778	19.03	
128 x 128	*Balloons*	*Flower*	31.96	0.9838	18.21	bior1.1
		Handwriting	32.4	0.9841	18.42	
		Pasta	32.0	0.9840	18.40	
	TigerPounce	*Flower*	32.69	0.9995	18.36	rbio2.8
		Handwriting	35.23	0.9995	18.56	
		Pasta	34.37	0.9995	18.56	
	F15Large	*Flower*	40.24	0.9920	18.17	rbio2.8
		Handwriting	39.59	0.9958	18.55	
		Pasta	40.58	0.9942	18.55	
	Zebras	*Flower*	22.62	0.9793	18.45	coif5
		Handwriting	22.5	0.9793	18.45	
		Pasta	22.68	0.9793	18.45	

Table 5 This table presents results from **CF-DWT-FB-GAR**. Highest PSNR and Capacity values are emphasized in a bold font.

Block size	Cover	Secret	PSNR	SSIM	Capacity	Wavelet
32 x 32	*Balloons*	*Flower*	32.18	0.9898	18.52	bior1.1
		Handwriting	32.8	0.9843	18.58	
		Pasta	32.35	0.9842	18.58	
	TigerPounce	*Flower*	33.08	0.9993	**21.40**	coif5
		Handwriting	33.2	0.9995	19.0	
		Pasta	32.3	0.9995	19.0	
	F15Large	*Flower*	**42.08**	0.9912	18.40	db7
		Handwriting	**42.44**	0.9923	18.83	
		Pasta	42.8	0.9911	18.82	
	Zebras	*Flower*	24.2	0.9778	18.5	db7
		Handwriting	24.2	0.9787	18.5	
		Pasta	24.2	0.9790	18.5	
64 x 64	*Balloons*	*Flower*	33.01	0.9835	18.53	bior1.1
		Handwriting	33.5	0.9880	19.1	
		Pasta	32.54	0.9579	16.22	
	TigerPounce	*Flower*	33.64	0.9995	18.1	bior4.4
		Handwriting	33.4	0.9994	19.4	
		Pasta	33.4	0.9994	19.4	
	F15Large	*Flower*	**41.60**	0.9936	18.31	db7
		Handwriting	**43.03**	0.9951	18.86	
		Pasta	40.6	0.9931	18.86	
	Zebras	*Flower*	22.74	0.9785	19.54	coif5
		Handwriting	22.62	0.9785	19.54	
		Pasta	22.53	0.9785	19.54	
128 x 128	*Balloons*	*Flower*	30.14	0.9827	19.52	bior1.1
		Handwriting	32.5	0.9838	18.6	
		Pasta	32.1	0.9879	18.8	
	TigerPounce	*Flower*	31.06	0.9993	20.51	rbio2.8
		Handwriting	35.23	0.9995	18.56	
		Pasta	34.37	0.9995	18.55	
	F15Large	*Flower*	**41.2**	0.9927	18.68	rbio2.8
		Handwriting	40.76	0.9956	19.15	
		Pasta	40.52	0.9943	19.15	
	Zebras	*Flower*	22.86	0.9795	19.17	coif5
		Handwriting	22.91	0.9795	19.17	
		Pasta	23.0	0.9794	19.17	

Table 6 This table presents results obtained from **QTAR**. Highest PSNR and Capacity values are emphasized in a bold font.

Block size	Threshold	Cover	Secret	PSNR	SSIM	Capacity
32 x 32	0.8	*Balloons*	*Flower*	**33.16**	0.9701	17.48
			Handwriting	31.63	0.9676	17.48
			Pasta	31.83	0.9676	17.48
	0.9	*TigerPounce*	*Flower*	28.85	0.9990	18.8
			Handwriting	31.48	0.9986	18.87
			Pasta	29.92	0.9991	18.87
	0.3	*F15Large*	*Flower*	24.74	0.9854	19.54
			Handwriting	27.23	0.9767	19.54
			Pasta	26.44	0.9820	19.54
	0.5	*Zebras*	*Flower*	23.22	0.9787	15.61
			Handwriting	22.51	0.9787	15.61
			Pasta	22.97	0.9787	15.61
64 x 64	0.8	*Balloons*	*Flower*	32.48	0.9856	17.97
			Handwriting	30.88	0.9849	17.96
			Pasta	31.14	0.9863	17.96
	0.9	*TigerPounce*	*Flower*	28.1	0.9969	19.37
			Handwriting	30.62	0.9950	19.37
			Pasta	29.31	0.9959	19.37
	0.3	*F15Large*	*Flower*	24.63	0.9853	19.97
			Handwriting	27.00	0.9787	19.96
			Pasta	26.25	0.9822	19.96
	0.5	*Zebras*	*Flower*	27.95	0.9871	15.61
			Handwriting	23.0	0.9787	15.61
			Pasta	23.0	0.9787	15.61
128 x 128	0.8	*Balloons*	*Flower*	31.81	0.9420	18.29
			Handwriting	30.2	0.9787	18.29
			Pasta	30.5	0.9787	18.29
	0.9	*TigerPounce*	*Flower*	27.23	0.9967	19.63
			Handwriting	29.97	0.9963	19.62
			Pasta	28.80	0.9787	19.63
	0.3	*F15Large*	*Flower*	27.21	0.9767	**21.01**
			Handwriting	26.13	0.9787	21.0
			Pasta	25.6	0.9787	21.0
	0.5	*Zebras*	*Flower*	22.9	0.9787	15.61
			Handwriting	23.0	0.9787	15.61
			Pasta	22.9	0.9787	15.61

Table 7 This table presents results obtained from **CF-QTAR**. Highest PSNR and Capacity values are emphasized in a bold font.

Block size	Threshold	Cover	Secret	PSNR	SSIM	Capacity
32 x 32	0.8	*Balloons*	*Flower*	**34.91**	0.9716	19.79
			Handwriting	34.1	0.9787	19.79
			Pasta	34.49	0.9787	19.79
	0.9	*TigerPounce*	*Flower*	32.88	0.9996	20.74
			Handwriting	33.22	0.9992	20.74
			Pasta	33.44	0.9993	20.74
	0.3	*F15Large*	*Flower*	28.22	0.9888	**22.61**
			Handwriting	27.35	0.9851	22.61
			Pasta	28.42	0.9868	22.61
	0.5	*Zebras*	*Flower*	28.27	0.9787	16.1
			Handwriting	27.51	0.9787	16.1
			Pasta	28.0	0.9787	16.1
64 x 64	0.8	*Balloons*	*Flower*	34.39	0.9934	20.0
			Handwriting	33.61	0.9987	20.0
			Pasta	34.1	0.9987	20.0
	0.9	*TigerPounce*	*Flower*	32.82	0.9996	20.83
			Handwriting	33.13	0.9992	20.83
			Pasta	32.54	0.9979	20.83
	0.3	*F15Large*	*Flower*	28.23	0.9875	**22.61**
			Handwriting	27.37	0.9844	22.61
			Pasta	28.44	0.9860	22.61
	0.5	*Zebras*	*Flower*	27.47	0.9787	16.1
			Handwriting	26.75	0.9787	16.1
			Pasta	27.19	0.9787	16.1
128 x 128	0.8	*Balloons*	*Flower*	34.23	0.9752	19.54
			Handwriting	33.34	0.9787	19.54
			Pasta	33.85	0.9787	19.54
	0.9	*TigerPounce*	*Flower*	32.54	0.9995	20.83
			Handwriting	33.22	0.9992	20.83
			Pasta	33.44	0.9994	20.83
	0.3	*F15Large*	*Flower*	28.16	0.9515	22.52
			Handwriting	27.39	0.9869	22.52
			Pasta	28.40	0.9891	22.52
	0.5	*Zebras*	*Flower*	27.45	0.9787	16.1
			Handwriting	26.72	0.9767	16.1
			Pasta	27.16	0.9767	16.1

Table 8 This table presents results obtained from **DWT-QTAR**. Highest PSNR and Capacity values are emphasized in a bold font.

Block size	Threshold	Cover (Wavelet)	Secret	PSNR	SSIM	Capacity
32 x 32	0.8	*Balloons (bior1.1)*	*Flower*	31.4	0.9833	18.5
			Handwriting	31.47	0.9830	18.4
			Pasta	31.5	0.9839	18.4
	0.9	*TigerPounce* (rbio2.8)	*Flower*	32.48	0.9995	18.27
			Handwriting	26.71	0.9989	**20.71**
			Pasta	27.02	0.9989	20.71
	0.3	*F15Large (db7)*	*Flower*	39.94	0.9896	18.03
			Handwriting	**42.7**	0.9951	18.8
			Pasta	36.2	0.9897	20.7
	0.5	*Zebras (db7)*	*Flower*	23.16	0.9787	18.07
			Handwriting	23.1	0.9781	18.07
			Pasta	23.0	0.9781	18.1
64 x 64	0.8	*Balloons (bior1.1)*	*Flower*	28.8	0.9823	19.97
			Handwriting	30.73	0.9834	18.4
			Pasta	30.6	0.9835	18.4
	0.9	*TigerPounce (rbio2.8)*	*Flower*	31.83	**0.9994**	19.5
			Handwriting	30.5	0.9994	18.1
			Pasta	32.50	0.9994	18.72
	0.3	*F15Large (db7)*	*Flower*	37.5	0.9922	18.03
			Handwriting	35.84	0.9893	18.70
			Pasta	36.5	0.9919	18.72
	0.5	*Zebras (db7)*	*Flower*	22.70	0.9781	18.02
			Handwriting	22.2	0.9753	18.02
			Pasta	22.70	0.9781	18.02
128 x 128	0.8	*Balloons (bior1.1)*	*Flower*	32.17	0.9824	18.2
			Handwriting	31.21	0.9821	18.1
			Pasta	32.0	0.9800	18.2
	0.9	*TigerPounce (rbio2.8)*	*Flower*	31.7	0.9994	19.5
			Handwriting	32.45	0.9994	18.38
			Pasta	34.49	0.9994	18.38
	0.3	*F15Large (db7)*	*Flower*	41.05	0.9921	18.1
			Handwriting	42.30	0.9952	18.2
			Pasta	42.01	0.9941	18.2
	0.5	*Zebras (db7)*	*Flower*	22.56	0.9780	18.3
			Handwriting	22.7	0.9780	18.33
			Pasta	22.82	0.9787	18.33

Table 9 This table presents results obtained from **CF-DWT-QTAR**. Highest PSNR and Capacity values are emphasized in a bold font.

Block size	Threshold	Cover (Wavelet)	Secret	PSNR	SSIM	Capacity
32 x 32	0.8	*Balloons (bior1.1)*	*Flower*	31.4	0.9833	18.5
			Handwriting	32.47	0.9839	18.64
			Pasta	32.3	0.9839	18.65
	0.9	*TigerPounce (rbio2.8)*	*Flower*	33.80	0.9991	22.52
			Handwriting	30.8	0.9991	**22.53**
			Pasta	31.1	0.9991	22.53
	0.3	*F15Large (db7)*	*Flower*	37.8	0.9908	21.26
			Handwriting	37.85	0.9931	21.26
			Pasta	37.78	0.9906	21.2
	0.5	*Zebras (db7)*	*Flower*	27.95	0.9871	19.02
			Handwriting	22.62	0.9781	19.02
			Pasta	27.07	0.9830	19.02
64 x 64	0.8	*Balloons (bior1.1)*	*Flower*	28.1	0.9769	22.03
			Handwriting	31.19	0.9837	18.93
			Pasta	31.0	0.9837	18.93
	0.9	*TigerPounce (rbio2.8)*	*Flower*	30.68	0.9994	20.38
			Handwriting	32.52	**0.9995**	19.47
			Pasta	32.46	0.9995	19.47
	0.3	*F15Large (db7)*	*Flower*	35.83	0.9896	22.0
			Handwriting	35.16	0.9905	22.0
			Pasta	36.66	0.9896	22.0
	0.5	*Zebras (db7)*	*Flower*	27.95	0.9871	18.84
			Handwriting	22.59	0.9784	18.84
			Pasta	22.59	0.9783	18.84
128 x 128	0.8	*Balloons (bior1.1)*	*Flower*	30.07	0.9824	19.87
			Handwriting	32.30	0.9836	18.81
			Pasta	32.0	0.9836	18.82
	0.9	*TigerPounce (rbio2.8)*	*Flower*	31.06	0.9993	20.5
			Handwriting	34.49	0.9994	19.53
			Pasta	33.61	0.9994	19.53
	0.3	*F15Large (db7)*	*Flower*	41.11	0.9927	18.68
			Handwriting	**43.41**	0.9952	18.77
			Pasta	42.22	0.9922	18.77
	0.5	*Zebras (db7)*	*Flower*	27.95	0.9871	19.22
			Handwriting	23.0	0.9780	19.22
			Pasta	23.12	0.9779	19.22

Table 10 A comparison of the **CF-QTAR** scheme at various other thresholds which produce improved results for highly correlated cover images and using the "Flower" secret image for embedding. Highest PSNR, SSIM and Capacity values for each block size level are emphasized in a bold font.

Block size	Threshold	Cover	PSNR (dB)	SSIM	Capacity (bpp)
32 x 32	0.1	*Balloons*	35.00	0.9637	19.62
		TigerPounce	33.19	**0.9996**	20.39
		F15Large	28.71	0.9241	22.16
	0.5	*Balloons*	**35.02**	0.9654	**19.88**
		TigerPounce	33.18	**0.9996**	20.65
		F15Large	28.15	0.9229	**22.70**
	0.9	*Balloons*	34.70	0.9720	19.79
		TigerPounce	32.88	**0.9996**	20.74
		F15Large	28.64	0.9251	22.07
64 x 64	0.1	*Balloons*	34.37	0.9673	20.05
		TigerPounce	32.63	0.9994	20.83
		F15Large	28.45	0.9197	**22.43**
	0.5	*Balloons*	34.32	0.9716	20.05
		TigerPounce	33.03	0.9994	20.83
		F15Large	28.18	0.9205	**22.61**
	0.9	*Balloons*	34.42	0.9633	19.88
		TigerPounce	32.82	**0.9996**	20.83
		F15Large	28.65	0.9214	22.07
128 x 128	0.1	*Balloons*	34.00	0.9767	19.79
		TigerPounce	32.54	0.9995	20.83
		F15Large	28.15	0.9512	**22.52**
	0.5	*Balloons*	34.00	0.9767	19.79
		TigerPounce	32.54	0.9995	20.83
		F15Large	28.16	0.9515	**22.52**
	0.9	*Balloons*	34.2	0.9752	19.54
		TigerPounce	32.54	0.9995	20.83
		F15Large	28.65	0.9525	**22.07**

5.4 Robustness Test

In this section we examine the robustness of our Curve-Fitting embedding scheme when the stego image has been attacked by different types of additive noise as well as various geometric deformations.

First, we simulate tampering the stego image with an information loss attack by deleting a square region of different sizes. The results are shown in figure 18.

Next, we add Salt and Papper noise, and Speckle noise to the stego image. Our embedding scheme also successfully passed through the stretching and warping attacks which are considered as geometric attacks, albeit at higher levels of degradation to the extracted hidden image.

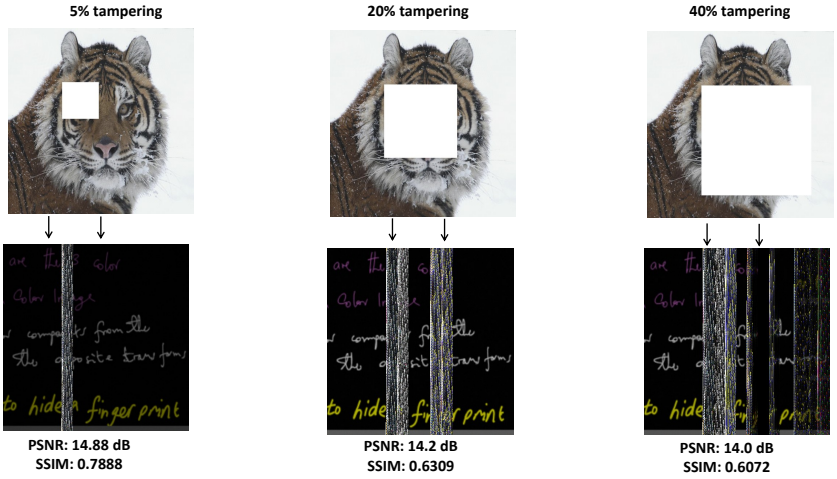

Fig. 18 Tampering a "Tigerface" stego image with a white square at different sizes to simulate data-loss attack. The extracted secret image is at an acceptably readable state for small amounts of data-loss.

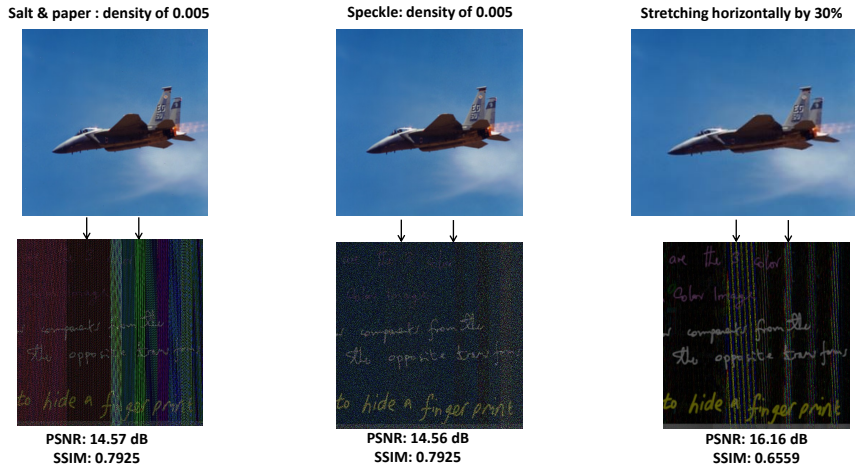

Fig. 19 Tampering with different types of degredations for a "Tigerface" stego image where the "Handwriting" secret image is embedded. From left to right; Salt and Pepper, Speckle, and Streching deformation attacks.

Figure 19 shows the robustness test using different types of noise applied as well as the stretching demormation attack. Figure 20 shows the extracted hidden image after degrading the stego image with a warping geometric attack at $20°, 30°, 40°$ degrees for the "Flower" and "Handwriting" secret images. The proposed method clearly comes short of extracting an acceptable hidden image evident by the low SSIM values.

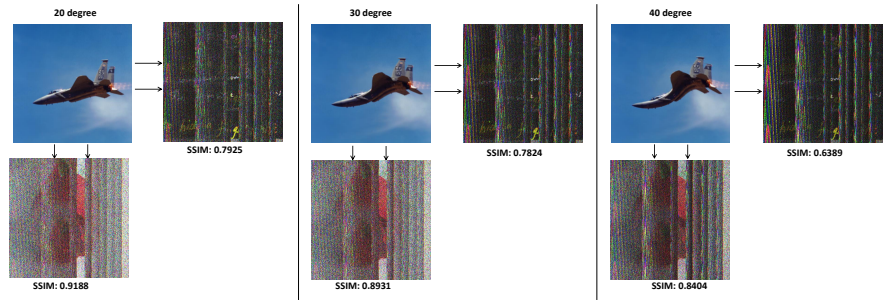

Fig. 20 Trying to recover after deforming the stego image with a warping attack at $20°, 30°, 40°$ degrees for the "Flower" and "Handwriting" secret images. The proposed method clearly comes short of extracting an acceptable hidden image evident by the low SSIM values.

6 CONCLUSIONS

This paper has introduced a Curve-Fitting (CF) approach to transform domain steganography schemes that demonstrated improvements in capacity as well as stego visual fidelity. This work complements previous work published in the literature [32–34] investigating the relationship between hiding capacity and stego image quality. This new methodology was implemented as the CF-FB-GAR, CF-QTAR, CF-DWT-FB-GAR, and CF-DWT-QTAR schemes and compared against the non-curve-fitted FB-GAR, QTAR, DWT-FB-GAR, and DWT-QTAR schemes. In CF-QTAR, the idea was to segment the cover image into blocks using the quad-tree segmentation algorithm thus forming statistically stationary regions of increasing sizes. The secret data is then hidden in the least significant areas of the DCT of each block by using a quantization step followed by piecewise linear curve fitting to three points bordering the full least significant DCT coefficient area. Experimental results and comparative evaluation have confirmed that although the embedding capacity has increased when using curve-fitting compared to earlier work, the perceptibility level has also improved over previous methods, thus breaking the traditional barrier that has confined the relationship between capacity and perceptibility to either higher capacities with reduced perceptual quality or higher perceptual quality albeit at the expense of lower capacities.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable suggestions that helped improve the original manuscript. This work was supported by the College of Graduate Studies and Research at the University of Sharjah.

References

1. Ahmed, N., Natarajan, T., Rao, K.: Discrete cosine transform. IEEE Trans. Computers **23**(1), 90–93 (1974)
2. Anderson, R.J., Petitcolas, F.A.: On the limits of steganography. IEEE Journal on selected areas in communications **16**(4), 474–481 (1998)
3. Bracamonte, J., Ansorge, M., Pellandini, F., Farine, P.A.: Low complexity image matching in the compressed domain by using the dct-phase. In: Proc. of the 6th COST, vol. 276, pp. 88–93 (2000)
4. Bracamonte, J., Ansorge, M., Pellandini, F., Farine, P.A.: Efficient compressed domain target image search and retrieval. In: Image and Video Retrieval, pp. 154–163. Springer (2005)
5. Brisbane, G., aini, R.S.N., Ogunbona, P.: High-capacity steganography using a shared colour palette. IEE Proc., Vis. Image Signal Process. **152**(6), 787–792 (2005)
6. Chan, C.K., Cheng, L.: Hiding data in images by simple LSB substitution. Pattern Recognition **37**, 469–474 (2004)
7. Chang, C.C., Chen, T.S., Chung, L.Z.: A steganographic method based upon jpeg and quantization table modification. Information Sciences **141**(1), 123–138 (2002)
8. Chen, B., Wornell, G.: Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Trans. Information Theory **47**(4), 1423–1443 (2001)
9. Cole, E.: Hiding in Plain Sight: Steganography and the Art of Covert Communication, 1 edn. John Wiley & Sons, Inc., New York, NY, USA (2003)
10. Ebrahimpour-Komleh, H., Chandran, V., Sridharan, S.: Face recognition using fractal codes. In: Image Processing, 2001. Proceedings. 2001 International Conference on, vol. 3, pp. 58–61. IEEE (2001)
11. El Safy, R., Zayed, H., El Dessouki, A.: An adaptive steganographic technique based on integer wavelet transform. In: Networking and Media Convergence, 2009. ICNM 2009. International Conference on, pp. 111–117. IEEE (2009)
12. Ibaida, A., Khalil, I.: Wavelet-based ecg steganography for protecting patient confidential information in point-of-care systems. IEEE Transactions on biomedical engineering **60**(12), 3322–3330 (2013)
13. IEC, I.: Information technology-digital compression and coding of continuous-tone still images: Requirements and guidelines. Standard, ISO IEC pp. 10,918–1 (1994)
14. Jain, A., Uludag, U., Hsu, R.: Hiding a face in a fingerprint image. In: Proc. of the International Conference on Pattern Recognition (ICPR). Quebec City, Canada (2002)
15. Lee, Y., Chen, L.: High capacity image steganographic model. IEE Proc., Vis. Image Signal Process. **147**(3), 288–294 (2000)
16. Lee, Y.K., Chen, L.H.: High capacity image steganographic model. IEE Proceedings-Vision, Image and Signal Processing **147**(3), 288–294 (2000)
17. Leng, C.K., Labadin, J., Juan, S.F.S.: Steganography: Dct coefficients reparation technique in jpeg image. JDCTA **2**(2), 35–41 (2008)
18. Lin, C.C., Shiu, P.F.: High capacity data hiding scheme for dct-based images. Journal of Information Hiding and Multimedia Signal Processing **1**(3), 220–240 (2010)
19. Lu, P., Luo, X., Tang, Q., Shen, L.: An improved sample pairs method for detection of lsb embedding. In: International Workshop on Information Hiding, pp. 116–127. Springer (2004)
20. Marvel, L.M., Charles G. Boncelet, J., Retter, C.T.: Spread spectrum image steganography. IEEE Trans. Image Processing **8**(8), 1075–1083 (1999)
21. Nozaki, K., Niimi, M., Eason, R.O., Kawaguchi, E.: A large capacity steganography using color bmp images. In: ACCV '98: Proceedings of the Third Asian Conference on Computer Vision-Volume I, pp. 112–119. Springer-Verlag, London, UK (1998)
22. Pavlidis, G., Tsompanopoulos, A., Papamarkos, N., Chamzas, C.: Jpeg2000 over noisy communication channels thorough evaluation and cost analysis. Signal Processing: Image Communication **18**(6), 497–514 (2003)
23. Qazanfari, K., Safabakhsh, R.: High-capacity method for hiding data in the discrete cosine transform domain. Journal of Electronic Imaging **22**(4), 043,009–043,009 (2013)

24. Qin, C., Chang, C.C., Chiu, Y.P.: A novel joint data-hiding and compression scheme based on SMVQ and image inpainting. Image Processing, IEEE Transactions on **23**(3), 969–978 (2014)
25. Qin, C., Chang, C.C., Hsu, T.J.: Reversible data hiding scheme based on exploiting modification direction with two steganographic images. Multimedia Tools and Applications **74**(15), 5861–5872 (2015)
26. Qin, C., Chang, C.C., Huang, Y.H., Liao, L.T.: An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism. Circuits and Systems for Video Technology, IEEE Transactions on **23**(7), 1109–1118 (2013)
27. Qin, C., Zhang, X.: Effective reversible data hiding in encrypted image with privacy protection for image content. Journal of Visual Communication and Image Representation **31**, 154–164 (2015)
28. Rabie, T.: Frequency-domain data hiding based on the matryoshka principle. Special Issue on Advances in Video Processing and Security Analysis for Multimedia Communications, International Journal of Advanced Media and Communication **1**(3), 298–312 (2007)
29. Rabie, T.: Data secrecy: An FFT approach. Advanced Techniques in Multimedia Watermarking: Image, Video and Audio Applications pp. 21–35 (2010). DOI: 10.4018/978-1-61520-903-3.ch002
30. Rabie, T.: High-capacity steganography. In: 6th International Congress on Image and Signal Processing (CISP), vol. 2, pp. 858–863 (2013)
31. Rabie, T.: Color-secure digital image compression. Multimedia Tools and Applications pp. DOI:10.1007/s11,042–016–3942–9 (2016)
32. Rabie, T., Kamel, I.: On the embedding limits of the discrete cosine transform. Multimedia Tools and Applications **74**(8), DOI:10.1007/s11,042–015–2557–x (2015)
33. Rabie, T., Kamel, I.: High-capacity steganography: A global-adaptive-region discrete cosine transform approach. Multimedia Tools and Applications **75**(2), DOI:10.1007/s11,042–016–3301–x (2016)
34. Rabie, T., Kamel, I.: Toward optimal embedding capacity for transform domain steganography: A quad-tree adaptive-region approach. Multimedia Tools and Applications **75**(7), DOI: 10.1007/s11,042–016–3501–4 (2016)
35. Raja, K., Chowdary, C., Venugopal, K., Patnaik, L.: A secure image steganography using lsb, dct and compression techniques on raw images. In: 2005 3rd International Conference on Intelligent Sensing and Information Processing, pp. 170–176. IEEE (2005)
36. Raja, K., Venugopal, K., Patnaik, L., et al.: High capacity lossless secure image steganography using wavelets. In: 2006 International Conference on Advanced Computing and Communications, pp. 230–235. IEEE (2006)
37. Rao, K., Yip, P.: Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press, ISBN 0-12-580203-X, Boston (1990)
38. Rodrigues, J., Rios, J., Puech, W., et al.: Ssb-4 system of steganography using bit 4. In: 5th International Workshop on Image Analysis for Multimedia Interactive Services (2004)
39. Roque, J.J., Minguet, J.M.: Slsb: Improving the steganographic algorithm lsb. In: WOSIS, pp. 57–66 (2009)
40. Solanki, K., Jacobsen, N., Madhow, U., Manjunath, B.S., Chandrasekaran, S.: Robust image-adaptive data hiding using erasure and error correction. IEEE Trans. Image Processing **13**(12), 1627–1639 (2004)
41. Sumathi, C., Santanam, T., Umamaheswari, G.: A study of various steganographic techniques used for information hiding. arXiv preprint arXiv:1401.5561 (2014)
42. Wang, Z., Bovik, A.C.: Mean squared error: love it or leave it? a new look at signal fidelity measures. Signal Processing Magazine, IEEE **26**(1), 98–117 (2009)
43. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. Image Processing, IEEE Transactions on **13**(4), 600–612 (2004)
44. Yang, B., Schmucker, M., Funk, W., Busch, C., Sun, S.: Integer dct-based reversible watermarking for images using companding technique. Proc. SPIE 5306, Security, Steganography, and Watermarking of Multimedia Contents **6**(405) (2004)