Springer Multimedia Tools and Applications Journal manuscript No.
(will be inserted by the editor)

A New Region Aware Invisible Robust Blind Watermarking Approach

Umar Albalawi · Saraju P. Mohanty · Elias Kougianos

Received: 01 March 2015

Abstract The multimedia revolution has made a strong impact on our society. The explosive growth of Internet access to this digital information has generated new opportunities and challenges. The ease of editing and duplicating information in the digital domain has created copyright protection concerns for content providers. Various schemes to embed secondary data in digital media have been investigated to preserve copyright and to discourage unauthorized duplication. In order to achieve the required level of protection to digital information, digital watermarking is a viable solution. This paper proposes a novel invisible watermarking scheme: a Discrete Cosine Transform (DCT) domain based watermark embedding and blind extraction algorithm for copyright protection of color images. The proposed algorithm is optimized in terms of robustness, computational load, and quality of the image. Testing of the proposed watermarking scheme's robustness and security via different benchmarks as well as comparison with a Discrete Wavelet Transform (DWT) approach proves its resilience to digital attacks. The results show that the Peak Signal to Noise Ratio (PSNR) value obtained under no attack conditions in the proposed algorithm is above 41.81 dB, which is higher than most of the existing watermark algorithms. The results from the detector's response also show that the proposed algorithm has a better security performance than most of the existing algorithms. The architecture for hardware implementation of the proposed algorithm is also presented and the results from the architecture show a PSNR value of 44.37 dB. This verifies that the presented algorithm is more effective than exist-

Computer Science and Engineering, University of North Texas, Denton, TX 76203.
Tel.: +1 940-565-3276
Fax: +1 940-565-2799
E-mail: UmarAlbalawi@my.unt.edu

Computer Science and Engineering, University of North Texas, Denton, TX 76203.
Tel.: +1 940-565-3276
Fax: +1 940-565-2799
E-mail: saraju.mohanty@unt.edu

Engineering Technology, University of North Texas, Denton, TX 76203.
Tel.: +1 940-891-6708
Fax: +1 940-565-2666
E-mail: elias.kougianos@unt.edu

ing algorithms such as image adaptive watermarking, image adaptive watermark creation, zerotree wavelet, and 9/7 biorthogonal wavelet lifting.

Keywords Digital watermarking; blind watermarking; invisible watermarking; robust watermarking; DCT domain; DWT domain

1 Introduction

The Internet revolution has made very easy the distribution, transmission, and access of multi-media files such as pictures, videos, distance education, etc. This digital content can be accessed anywhere, anytime through the Internet, thus enabling very easy distribution and increased share in the global market. However, this easy access of digital content has brought many issues along with it. Manipulation, unauthorized access, and illegal distribution of digital content through the use of inexpensive tools have caused industries like music and movies to incur losses. The authors, distributors, and creators are losing revenue due to illegal access and distribution of their original works. The laws to prevent these illegal acts do not appear to be effective enough to deter law breakers. Digital Right Management (DRM) addresses these concerns about protection of intellectual property (IP).

In this paper, an attempt is made to protect an image against unauthorized modification and false ownership claims. A technique known as digital watermarking is employed. Digital watermarking is a method where information about the image or owner is embedded in the image itself in such a way that an unauthorized user cannot recover an original copy of the image while an authorized user can, after necessary processing of the image. Depending upon the application requirements, one can employ different types of watermarking. Figure 1 shows a general classification of watermarking.

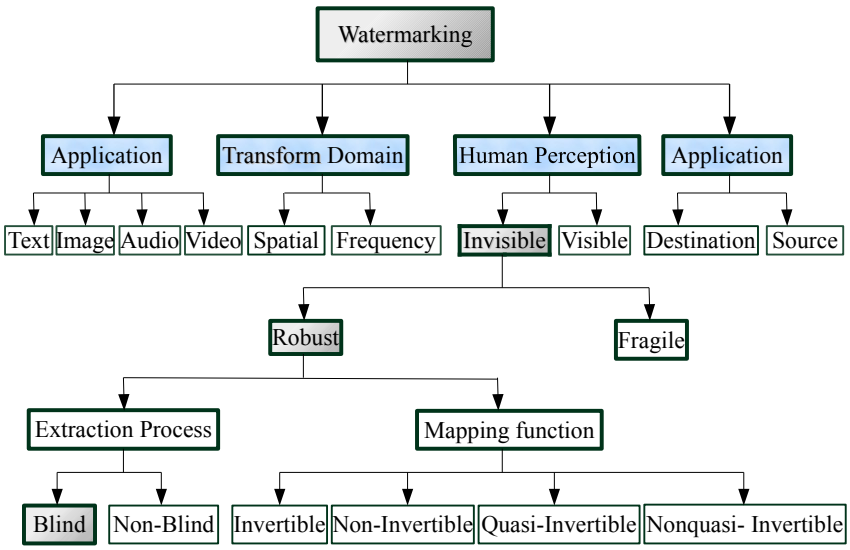


Fig. 1 Different types of digital watermarking. The type of watermark used in this work is denoted by the grey rectangles.

Each of these watermarking schemes has its own applications and usefulness [1,2,3]. Visible watermarking is used in applications like broadcasting, where the broadcaster's logo is embedded in the video. Invisible robust watermarking is used when the ownership needs to be maintained. Invisible fragile watermarking is embedded in such a way that any modification done to the watermarked image will destroy the watermark. Such a watermarking scheme can be very useful in applications like news reporting to claim the originality of the captured scene. Dual watermarking is used where both visible and invisible watermarks are used. The invisible watermark is used as a secondary back up.

Source-based watermarking is used in applications where the unique watermark of the owner is introduced in all copies of the digital content so that any alteration can be easily recognized. Destination based watermarking is used when the distributed copies are given a unique watermark identifier of the buyer and any infringement made by the buyer, like illegal distribution or alteration, can be identified based on the watermark recovered from the watermarked copy.

Depending on the requirements of the watermark, such as speed, cost, and reliability, the watermarking process can be done either in the spatial/time or in the frequency domains. Spatial/time domain watermarking is faster and easier but is not as robust as frequency domain watermarking. Since frequency domain watermarking is very robust to attacks, it is mostly employed where security is the prime concern.

Each of these watermarking schemes is equally important and applicable. However, the choice of watermarking depends on the application. Each of these watermark algorithms should follow some common criteria such as:

1. The watermark should not affect the intelligibility of the host digital content (image, audio, or video) in the case of a visible watermark or should not be visible in the case of an invisible watermark.
2. In the case of a non-fragile watermark, it should be resistant to most common intentional or unintentional attacks like noise, cropping, scaling, and filtering.
3. The original digital content should be recoverable from the watermarked content.
4. The process should not be very slow, complex or expensive.
5. It should be possible to detect the watermark with very low probability of false detection.
6. The presence of the watermark should not be obvious to anyone.

As can be seen from these criteria, there are trade offs among quality, robustness, and speed. Robustness asks for more alteration in the host content, whereas quality and speed require few pixels to be modified. So, a very careful design is a must in any type of watermarking algorithm.

The organization of the paper is as follows. Notations used in this paper are shown in Table 1. Section 2 describes the contributions of this paper. Section 3 describes related research in the field of invisible watermarking for both blind and non-blind algorithms. In Section 4, the proposed invisible watermarking algorithm for insertion and extraction is discussed in detail. Section 5 discusses experimental results on various test images and their detector's responses. Section 6 shows the proposed Simulink® model for the algorithm. Conclusions and future directions of this research are discussed in Section 7.

2 Contributions of this Paper

A schematic overview of the proposed watermarking algorithm is shown in Figure 2. The host color image is first converted to YCbCr color space from RGB. The actual processing is

Table 1 Notation used to describe the proposed watermarking algorithm.

Notation	Description
$\mathbf{O}$	Original color image
$\mathbf{Y}$	Y-component after conversion of $\mathbf{O}$ from RGB to YCbCr
$\mathbf{A}$	Watermark: pseudo random numbers with zero mean and unit variance
$\mathbf{O}'$	Watermarked image, also a color image
$\mathbf{O}^*$	Distorted image
$\mathbf{Y}'$	After IDCT from $\mathbf{Y}$
$\mathbf{Y}^*$	At detector, from processing $\mathbf{Y}'$
N $\times$ N	Dimensions of the test image $\mathbf{O}$ and corresponding watermarked image $\mathbf{O}'$
σ	Watermark detection ratio
Th_σ	Watermark detection threshold
α	Scaling constant, watermark strength factor

done only on the luminance (Y) component. From this Y component of the image, the center quarter of the image, where the watermark will be embedded, is extracted. The motivation behind the use of the center quarter of the host image for watermark placement is due to reduced computation load and increased robustness requirements. A pseudo-random number sequence with zero mean and unit variance is used as the watermark in the host image. After inserting the watermark in the host image, the Cb and Cr components are added to the processed Y component thus producing the final watermarked color image.

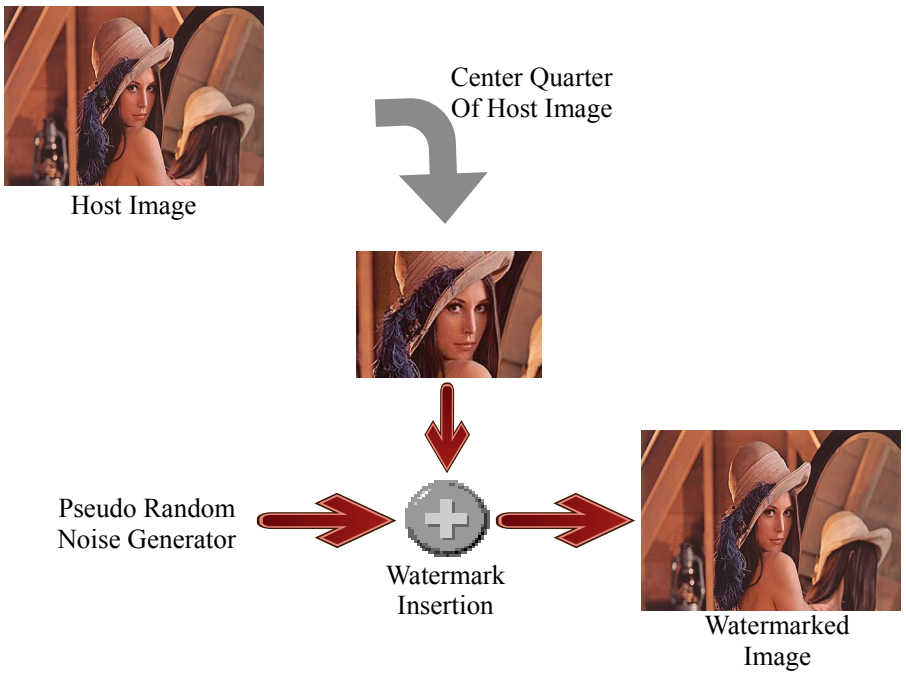


Fig. 2 Schematic overview of the proposed watermarking scheme.

This paper's contributions to the invisible-robust-blind watermarking algorithm are summarized as follows:

1. Optimization of robustness, quality, and computational load. Since the center portion of the image contains the main information about the image, watermarking insertion at this center quarter increase robustness. Any attempt to remove the watermark will result in degradation of image quality. Because the center quarter of the host image is only used for watermarking, it increases the overall image quality. The computation needed to insert the watermark is also greatly reduced due to the fact that only 25% of the host image is processed for watermarking. Thus, this method achieves the three contradictory requirements of the watermarking process, namely robustness, quality, and computational efficiency.
2. Increased watermarking insertion speed. Watermarking is done in the frequency domain using the Discrete Cosine Transform (DCT). However, instead of taking the complete DCT of the center quarter image, block-wise DCT is done where the block size is 8×8. This increases the computational speed.
3. Increased robustness. The insertion of the watermark is done in the mid-frequency of the image block so that the removal of high or low frequency components of the watermarked image by operations such as filtering, compression, and sampling does not significantly affect the watermark.
4. Increased quality. Energy compaction in low frequency components is maximal and any change in these low frequency components can degrade the image quality. Similarly, high frequency components contain edge information and other details, which are needed to preserve edge information in the image as the human eye is very sensitive to edges. Hence, the watermarking is done in the mid frequency so that any change can not be noticed by the human eye.
5. A hardware architecture for the proposed algorithm is implemented in Simulink®. The results from the simulation are compared with MATLAB®results, and indicate that Simulink®offers better PSNR than MATLAB®. This verifies that when the proposed model is implemented in hardware, it performs better than most existing algorithms.

3 Related Prior Research

Blind watermarking techniques are very useful as they do not require a host for authentication. Wong *et al.*[4] proposed three blind watermarking embedding techniques in order to produce high quality images, where the first technique (SWE) embeds a watermark with two secret keys using a spread spectrum technique, the second (MWE) uses different watermarks on different correlated secret keys, and the third (IWE) uses a watermark on a JPEG compressed image. Zong *et al.*[5] give a novel image watermarking approach based on rank method. In this approach, a secret key is used to randomly select a set of DCT coefficients. To insert the watermark bits, a rank-based embedding rule is used to modify this set.

Gunjal *et al.*[6] present a non-blind digital image watermarking technique where a comparative analysis of performance in the DWT and DWT-FWHT-SVD (DET-Fast Walse Hadamard Transform-Singular value decomposition) domains is given. Analyzing the experimental results show that DWT could not reach the performance of DWT-FWHT-SVD domain. Using the probability of total number of 0's and 1's in a single block, a new digital watermarking technique was proposed by Hawlader *et al.*[7]. A scrambled binary watermark is embedded into the image four times in four different locations for better security. During extraction, the total amount of 1 and 0 bits from each of the blocks of size 8×8 is

compared. Hamad *et al.*[8] propose an algorithm where a DNA-encoded watermark is complemented and hid in the third level resolution of the wavelet decomposition of the true color image. The invisible image embedding is enhanced by applying the quantization operation. Communication in an optical camera and its capability for invisible watermarking schemes are addressed in [9]. The invisible elements of the image are changed and the digital data are embedded for communication. [10] presents a method where the color image is embedded as an invisible watermark into a 3D image. This paper deals with the security aspects of 3D anaglyph images. DCT domain using the Quantization Index Modulation - Dither Modulation (QIM-DM) scheme is used for this insertion in order to achieve imperceptibility of watermark. Ghosh *et al.*[11] propose an algorithm with highest immunity to attacks and retain the robustness. Non-blind watermarking was implemented, which exploits the high level of robustness and results in perfect reconstruction every time. Kaur and Lal [12] present an algorithm which is based on both redundant second generation wavelet packet transform and fast haar wavelet transform. The cover image is decomposed according to the size of watermark. The algorithm presented in this paper aims at embedding the watermark information into the media cover to maintain authentication. Pushpalatha [13] proposes an effective algorithm designed for enhancing the quality of image using a 5/3 2D Lift DWT based technique for the investigation of the effect of PSNR for digital images. The watermark is generated as a pseudo random sequence in [14]. A comprehensive sensing scenario was assumed which affects the robustness of the watermark. Detection of the watermark in such situations is the main aim in [14]. Pathak *et al.*[15] present a watermarking technique in medical images where the difficulty resides in the measure of perceptual distortion. The algorithm presented in [15] offers a way to insert the watermark in images using SVD in DWT domain. To enhance the transmission security, encryption is used. A comparative analysis of watermarking techniques is presented in [16]. Simulations are conducted for PSNR using the LSB, DCT and DWT techniques. The LSB technique gives out a better PSNR where the DCT gives a better Normalized Coefficient (NC) value.

Miller *et al.*[17] presented a robust image watermark embedding 1380 bits of information in an image where the robustness is maintained by minimizing the perceptual distance. A modified trellis code with an iterative method was used in embedding. Liu *et al.*[18] presented an adaptive blind watermarking technique where two levels of wavelet coefficients are used to embed a watermark and a generalized Gaussian distribution is used to detect the statistical model for wavelet coefficient detection. Guannan *et al.*[19] proposed an adaptive block-based blind watermarking using the DWT where the binary image is embedded in selected sub-bands after analyzing the coefficient characteristics. Erhu and Fan [20] showed that image robustness against attacks can be increased by applying an integer wavelet transform where each block is decomposed into three levels for embedding the watermark. Yu [21] proposed a blind wavelet watermarking algorithm wherein the changes in sign of the DWT coefficients indicate the watermarking bits. Choi and Seo [22] proposed a statistical watermarking algorithm based on the Human Visual System (HVS), where each bit of the watermark image is placed in a spreading pattern in the DCT coefficients and the detection is done by the Estimated Weight Binary Hypothesis Test (EWBHT). Zhi-Bo *et al.*[23] proposed blind watermarking based on a 9/7 bi-orthogonal wavelet lifting transform where the extraction process is done from the low frequency domain. Yen *et al.*[24] proposed a 3-level wavelet transform based watermarking algorithm where the image is resolved into 10 sub-bands and the noise watermark derived from the binary watermark is used to substitute the third and fourth bit of the absolute values of the coefficients. Qiao *et al.*[25] presented an adaptive blind watermarking algorithm wherein the watermark is embedded in the DC component of the image based on the Human Visual System (HVS) model. However the

algorithm also shows that the watermark can be embedded in the mid-frequency range to withstand JPEG compression as well as attacks .

A non-blind watermarking algorithm needs a host image for the watermark extraction. Khalfallah *et al.*[26] proposed a better robust and imperceptible watermarking algorithm where the image is transformed into a multi-resolution field and the appropriate coefficients are selected for watermarking. An adaptive embedding strength is provided by two terms, where one of them is fixed and the other is variable chosen to reduce computational error. Yongliang *et al.*[27] proposed a public key encryption scheme called Rabin cryptosystem to provide high security against attackers. Safabakhsh *et al.*[28] presented a two dimensional DWT based watermarking algorithm for gray level images where a watermark is embedded in high pass wavelet coefficients of the host image. To obtain better detection algorithm, it is based on entropy and the HVS model to select watermarked coefficients from wavelet coefficients. Ganic and Eskicioglu [29] proposed a hybrid scheme based on 2-D DWT and SVD to increase the robustness against image compression schemes, histogram equalization, and geometric attacks. The watermark is embedded in low and high frequencies, and even the LL band is modified. Piper *et al.*[30] proposed a spread spectrum image watermarking algorithm using characteristics of the HVS. A constant embedding strength is used and the detection algorithm is based on the textured region so as to achieve quality scalability without compromising resolution scalability. Terzija *et al.*[31] proposed a complex wavelet transformed watermarking algorithm using error correction code to increase robustness. The watermark embedding is done in the spatial domain. Zaboli and Moin [32] developed an entropy based watermarking algorithm using HVS characteristics in the contourlet domain. Scrambling of the pseudo-random number sequence was done to increase the performance of watermark detection in the extraction process and the quality is improved by increasing the level of decomposition. Lavoue *et al.*[33] presented frequency domain watermarking using error correcting codes and modulation techniques to address the trade-off between redundancy and imperceptibility.

4 Proposed Region-Aware Blind Watermarking Algorithm

A robust invisible watermarking algorithm with blind extraction is presented in this section. The proposed algorithm first converts any color image from the RGB to the YCbCr color space. The advantage of this color space is that it takes human perception into account and enables the process to take place in the luminance part (i.e. the Y component of the image) and not the chrominance part (i.e. the Cb and Cr components). This improves the computational efficiency as only one color component has to be processed.

4.1 Insertion Algorithm

Watermarking insertion is done on a color image of size $N \times N$ based on the procedure proposed by Mohanty *et al.*in [34]. As a first step of insertion, the color image is transformed from the RBG space to the YCbCr space and only the Y component is considered for further processing. The Y image component is divided into an equal number of 8×8 blocks and DCT is performed on each block. Since the center portion of the image is the focus of attention from the viewers' point of view, the watermark will be embedded in the center quarter of the image. Since only this 25% of the image area is used for watermarking, computational efficiency as well as overall image quality are increased. Furthermore, if attackers try

to remove/destroy the watermark for this center portion, then the image quality will degrade. However, special care has to be taken when embedding a watermark in this center region, and any significant change in the host image at this region due to the watermark can degrade the quality of the image or make the viewer aware of the presence of watermarks.

The selection of suitable DCT coefficients for watermarking is very important. The low frequency coefficients contain much of the signal energy, and the human eye is also more sensitive to these frequencies as compared to high frequency coefficients. On the other hand, the high frequency coefficients contain edge information and details. Any change in low frequency coefficients due to watermarking may degrade the quality of the image and the watermark implemented in high frequency coefficients may be easily altered by attacks (intentional or unintentional) such as data compression, low pass filtering, and sub-sampling. Hence, it is best to select mid frequencies for watermark insertion to maintain the robustness to attacks and the quality of the watermarked image. Keeping this in mind, four mid-frequency coefficients are chosen as $C_{4,1}$, $C_{3,2}$, $C_{2,3}$, and $C_{1,4}$ (in Fig. 3 they are labeled 19, 18, 17 and 16) from each block in the center quarter of the image [4,17,35]. Through these coefficients, a vector R of size K is generated where K is the number of 8×8 blocks in the center quarter of the image:

$$R = \{r_{1,i}, r_{2,i}, r_{3,i}, r_{4,i}, \cdots, r_{1,K}, r_{2,K}, r_{3,K}, \cdots, r_{4,K}\}, \tag{1}$$

where $r_{x,y}$ is the coefficient of the selected block y.

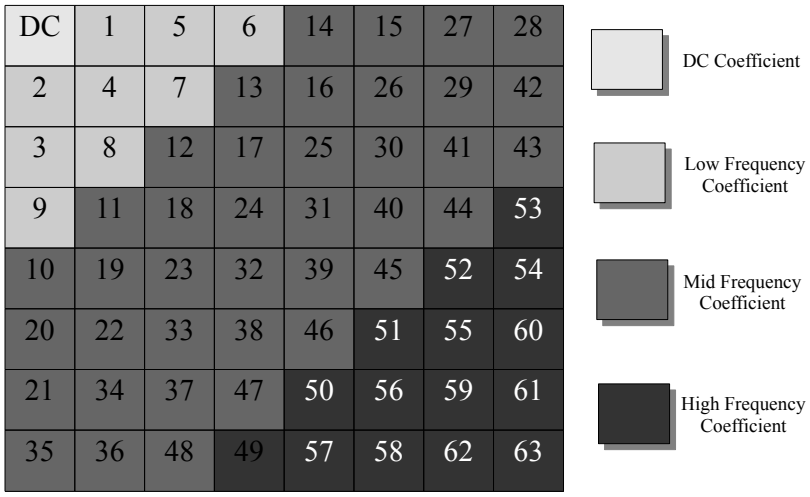


Fig. 3 DCT coefficient numbering scheme for an 8×8 block.

A pseudo random sequence is chosen from 1000 pseudo random sequences of size $4 \times K$ and is then used as the watermark represented as:

$$A = \{a_1, a_2, a_3, \cdots, a_{4\times K}\}, \tag{2}$$

where every element is of zero mean and unit variance. The watermark A is to be inserted into the DCT coefficients of the image of vector R according to:

$$r_i^{'} = r_i + \alpha \mid r_i \mid a_i, \tag{3}$$

for $i = 1, 2, \cdots, 4 \times K$ and α is a scaling constant, which is used to determine the watermark strength. Small values of α can make the watermark vulnerable to modification and also make it difficult to extract and detect during the detection/extraction stage. Similarly, large values of α can make the watermark visible. So an optimum choice of this scaling constant is necessary [22]. This process forms a new vector $R^{'}$ given by:

$$R^{'} = \{r_1^{'}, r_2^{'}, r_3^{'}, \cdots, r_{4\times K}^{'}\}, \tag{4}$$

which is of the same size as vector R. These new values of $R^{'}$ are reinserted into the DCT coefficients of the corresponding blocks, and then the block-wise inverse discrete cosine transform (IDCT) is applied. This gives the modified $Y^{'}$ component in the spatial domain. The Cb and Cr components of the host image are then concatenated with $Y^{'}$ in order to obtain the color image. This image is finally converted to the RGB space, and the resulting image is the watermarked image, $O^{'}$. The complete insertion and detection insertion process is shown in Fig. 4.

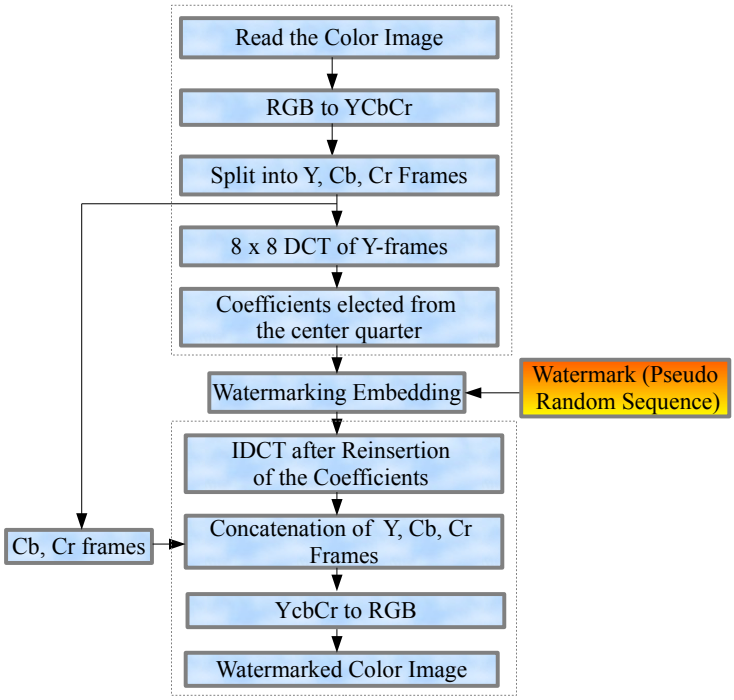


Fig. 4 Proposed watermark insertion algorithm.

4.2 Detection Algorithm

The essence of the blind algorithm is the non-availability of original image information at the detector side. This implies that when detecting for the presence of the watermark, no

information about the original image is used at any stage. This overcomes security issues, minimizes the memory size, extra circuitry, and delay for hardware implementation. During the watermarking process, the original $Y^{'}$CbCr has been converted to Y^{*}CbCr. There may be some corruption, noise, and distortion that could happen to the watermarked image due to transmission, attacks, and processing of the image. So this $Y^{'}$CbCr image (or $O^{'}$) is changed to image Y^{*}CbCr (or O^{*}). The DCT of O^{*} is then taken in a block-by-block manner where the block size is 8. The central quarter of the image is identified in order to extract the coefficients from which the watermark will be extracted. The same mid frequency coefficients have to be selected that have been considered in the watermark insertion process, as these sets of coefficients give information about the presence of watermark. From the chosen coefficients, a vector R^{*} of size $4 \times K$ is generated in order to provide information:

$$R^{*} = \{r_1^{*}, r_2^{*}, r_3^{*}, \cdots, r_{4\times K}^{*}\}. \tag{5}$$

To determine the presence of a watermark in image O^{*}, a correlation coefficient σ is derived according to the watermark A, which is inserted into the DCT coefficient of the image of vector R in Eqn. 3. The correlation coefficient σ is defined to compute the correlation between the extracted coefficients R^{*} and the watermark A itself using the formula:

$$\sigma = \frac{AR^{*}}{K} = \sum_{i=1}^{K} a_i r_i^{*}. \tag{6}$$

The ideal situation where the watermark is not corrupted occurs when:

$$r_i^{*} = r_i^{'} = r_i + \alpha \mid r_i \mid x_i, \tag{7}$$

where X is the vector of DCT coefficients. In that case, the correlation σ will be:

$$\sigma = \frac{1}{K} = \sum_{i=1}^{K} r_i a_i + \alpha \mid r_i \mid x_i a_i. \tag{8}$$

When A and X are matched the correlation σ will be:

$$\sigma = \frac{1}{K} = \sum_{i=1}^{K} r_i a_i + \alpha \mid r_i \mid a_i^2. \tag{9}$$

Assuming zero means of the vectors r_i's and x_i's, μ equals $\alpha\mu_{|r|}$ when $A = X$; otherwise, μ equals 0. A threshold Th_σ is defined so that by comparing the calculated σ with this threshold Th_σ, it is possible to determine the presence or absence of a watermark. In order to test the robustness of the proposed algorithm and increase the threshold, different values of threshold are selected. Starting from the value 2.0 proposed in [36], a trial and error method was followed by trying values of 1.8, 1.6, 1.4, 1.2, 1.0, and 0.8. The value 1.2 gives the best results over all test images considered. Thus, the threshold value has correspondingly increased to make the algorithm more immune to noise and at the same time less probable of making the wrong decision than the algorithm in [36]. This threshold is determined using the following expression:

$$Th_\sigma = \frac{\alpha}{1.2K} \sum_{i=1}^{K} \mid r_i^{*} \mid . \tag{10}$$

The decision is made based on the following: If $\sigma > Th_\sigma$ then a watermark is present and if $\sigma < Th_\sigma$ then a watermark is absent. This helps to make decisions about the authenticity of the image. The overall algorithm flow for watermark detection is presented in Fig. 5.

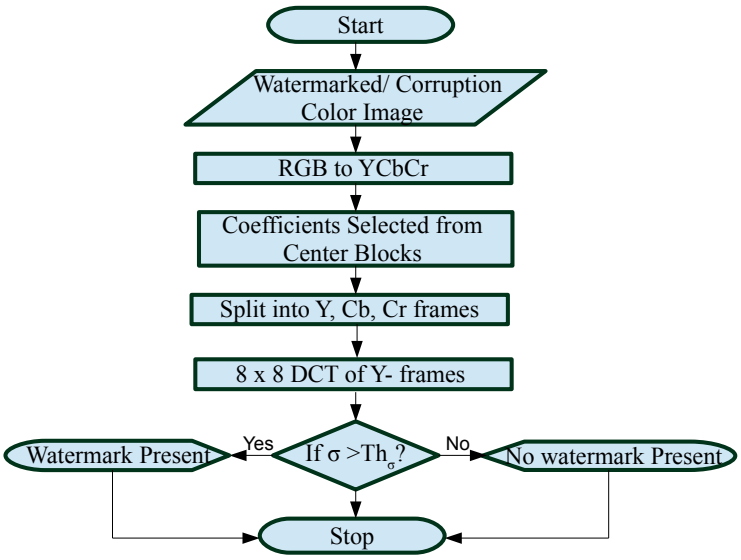


Fig. 5 Proposed watermark detection algorithm.

5 Experimental Results

The main objective of this paper is to determine robustness improvement without any loss in image quality. For this, the watermarking algorithm is implemented in MATLAB®and Simulink®. Implementing the algorithm in MATLAB®gives a better understanding of the low-level implementation while the Simulink®model provides a top-level functional and data-flow visualization, suitable for hardware implementation. Extensive testing of the proposed algorithm for several test images is done and the details are shown in this section.

5.1 Test of insertion and quality assurance

Several different images are selected randomly from a large set of images in Joint Picture Expert Graphics (.jpg) and Portable Network Graphics (.png) formats. The proposed invisible watermarking insertion algorithm is tested for this set of images as described below. The performance of the proposed algorithm increases with the size of the images. It can be seen from the results that the visual quality of the image is maintained so that the change in the image quality before and after watermarking cannot be perceived by the human eye. Selected results are presented in Figs. 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16.

5.2 Graphs of RMSE, PSNR, and SSIM vs. α

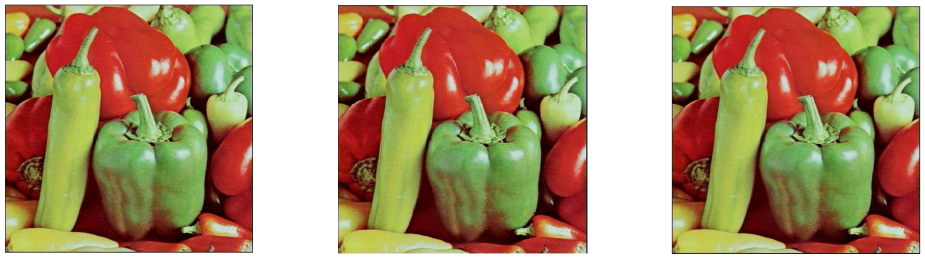
As discussed in the previous section, "α" dictates the perceived quality and strength of the watermark in the watermarked image. Thus, to measure the robustness and invisibility of the watermark, three performance measures are considered: the Root Mean Square Error (RMSE), the Peak Signal-to-Noise ratio (PSNR), and the Structural SIMilarity (SSIM).

(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 6 Watermarking of "Lena" (512×512).

(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 7 Watermarking of "Baboon" (225×225).



(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 8 Watermarking of "Pepper" (512×512).

(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 9 Watermarking of "F16" (512×512).

(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 10 Watermarking of "Forest" (256×256).

(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 11 Watermarking of "Wallpaper" (128×128).

(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 12 Watermarking of "Squirrel" (256×256).

(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 13 Watermarking of "Bear" (256×256).

(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 14 Watermarking of Google Map Image (256×256).

(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 15 Watermarking of "Resort" (256×256).

(a) Cover Image (b) Watermarked ($\alpha = 0.2$) (c) Watermarked ($\alpha = 0.65$)

Fig. 16 Watermarking of "Ice Climb" (512×512).

RMSE compares the extracted watermark $O^{'}$ to that of the stored original image O of size $m \times n$ and is given by the following expression:

$$RMSE = \frac{1}{\sqrt{mn}} \sum_{i=0}^{m-1} \sum_{j=0}^{n=1} || \, O(i,j) - O'(i,j) \, ||^2 \, . \tag{11}$$

PSNR is the ratio between the maximum possible energy of O (E_{max}) and the power of the corrupted image O' from watermarking and is given by the following expression:

$$PSNR = 20 \log_{10} \left(\frac{E_{max}}{RMSE} \right) . \tag{12}$$

SSIM aims to measure image resolution and viewing condition. Cover images are used as references and then the similarity between the watermarked images and references is

measured. SSIM correlates with human perception: luminance, contrast, and structure [37]:

$$SSIM(i,j) = L_{I,J}(i,j)C_{I,J}(i,j)S_{I,J}(i,j), \tag{13}$$

where L,C and S are the luminance, contrast, and structure, respectively, of the watermarked image. In these calculations, the value of α varies in the range [0.2 , 0.65]. Table 2, Table 3, and Table 4 show the RMSE, PSNR, and SSIM values obtained from MATLAB®for different values of α. From the results given in these three tables, it can be seen that the value for PSNR is maintained above 41.81 dB for all cases. As the visual quality of the watermarked images improves with larger values of PSNR, this result shows that the proposed algorithm maintains the quality of the watermarked images so that it is impossible for the human eye to visually detect the presence of any watermark in it. Higher values of PSNR also show how robust the algorithm is to different types of attack. For example the average PSNR for the "Pepper" image is 53.80 dB, which is the highest among all. The average of SSIM for the "Pepper" image is also the highest among all. This implies that the "Pepper" image is more robust with respect to noise.

Table 2 RMSE values of the analyzed images from MATLAB®simulations.

α	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65
Lena	0.58	0.71	0.84	0.96	1.07	1.18	1.29	1.38	1.48	1.56
Baboon	0.37	0.46	0.55	0.64	0.73	0.82	0.90	0.98	1.07	1.14
Pepper	0.27	0.34	0.40	0.47	0.53	0.59	0.65	0.70	0.76	0.81
F-16	0.52	0.65	0.77	0.89	1.00	1.11	1.21	1.31	1.40	1.48
Forest	0.54	0.67	0.81	0.93	1.06	1.18	1.30	1.42	1.53	1.63
Wallpaper	0.47	0.58	0.70	0.81	0.92	1.04	1.14	1.26	1.36	1.47
Squarrel	0.39	0.49	0.59	0.68	0.77	0.86	0.94	1.02	1.09	1.17
Bear	0.77	0.95	1.13	1.29	1.44	1.58	1.71	1.84	1.96	2.07
Google Map	0.71	0.87	1.03	1.18	1.33	1.47	1.61	1.73	1.85	1.96
Resort	0.57	0.71	0.85	0.99	1.12	1.25	1.37	1.49	1.60	1.71
Ice Climb	0.34	0.42	0.51	0.59	0.66	0.74	0.80	0.87	0.93	0.99

Table 3 PSNR values of the analyzed images from MATLAB®implementation.

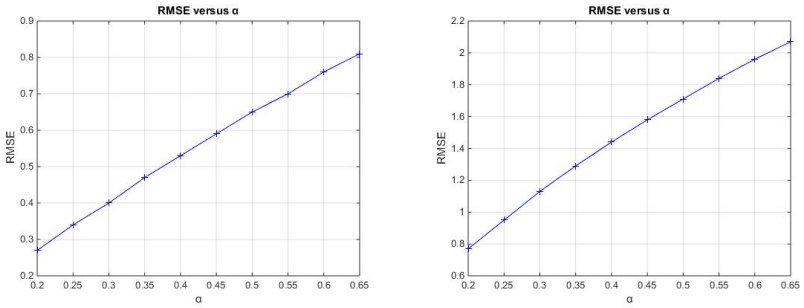
α	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65
Lena	52.90	51.11	49.69	48.51	47.51	46.67	45.95	45.31	44.75	44.25
Baboon	56.71	54.89	53.32	52.02	50.55	49.89	49.04	48.27	47.55	46.97
Pepper	59.53	57.61	56.09	54.78	53.69	52.73	51.92	51.18	50.52	49.94
F-16	53.81	51.90	50.38	49.14	48.09	47.23	46.48	45.81	45.23	44.73
Forest	53.41	51.55	49.73	48.73	47.63	46.69	45.83	45.10	44.44	43.87
Wallpaper	54.63	52.83	51.27	49.98	48.85	47.81	46.96	46.14	45.44	44.78
Squarrel	56.20	54.29	52.78	51.45	50.37	49.42	48.67	47.98	47.36	46.79
Bear	50.37	48.55	47.09	45.94	44.94	44.14	43.45	42.84	42.30	41.81
Google Map	51.11	49.32	47.87	46.66	45.65	44.77	44.02	43.35	42.79	42.28
Resort	52.95	51.08	49.50	48.21	47.14	46.19	45.38	44.67	44.04	43.49
Ice Climb	57.47	55.57	54.01	52.72	51.68	50.80	50.04	49.36	48.75	48.19

As mentioned, from Tables 2, 3, and 4, the average PSNR value for the "Pepper" image is maximum, 53.80 dB. Similarly, the minimum value for average PSNR is for the "Bear" image, which is 45.14 dB. The graphs of PSNR versus α, RMSE versus α, and SSIM versus α for the "Pepper" and "Bear" images are shown in Fig. (17(a)), (17(b)), (18(a)), (18(b)),

Table 4 SSIM values of the analyzed images from MATLAB®implementation.

α	**0.2**	**0.25**	**0.3**	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**
Lena	0.9985	0.9984	0.9983	0.9981	0.9977	0.9977	0.9974	0.9971	0.9969	0.9965
Baboon	0.9954	0.9954	0.9953	0.9953	0.9952	0.9952	0.9951	0.9950	0.9950	0.9949
Pepper	0.9996	0.9996	0.9996	0.9996	0.9995	0.9995	0.9995	0.9994	0.9994	0.9994
F-16	0.9771	0.9770	0.9769	0.9768	0.9768	0.9766	0.9765	0.9763	0.9762	0.9760
Forest	0.9909	0.9909	0.9908	0.9907	0.9906	0.9905	0.9903	0.9902	0.9900	0.9899
Wallpaper	0.9544	0.9543	0.9541	0.9540	0.9538	0.9536	0.9532	0.9532	0.9529	0.9527
Squarrel	0.9602	0.9601	0.9600	0.9599	0.9597	0.9595	0.9594	0.9591	0.9589	0.9587
Bear	0.9442	0.9442	0.9441	0.9440	0.9439	0.9437	0.9435	0.9434	0.9432	0.9430
Google Map	0.9994	0.9991	0.9988	0.9984	0.9981	0.9977	0.9968	0.9963	0.9958	0.9952
Resort	0.9903	0.9902	0.9901	0.9900	0.9899	0.9897	0.9895	0.9894	0.9892	0.9889
Ice Climb	0.9774	0.9773	0.9771	0.9770	0.9768	0.9766	0.9764	0.9760	0.9757	0.9751

(19(a)), and (19(b)). All other images have the PSNR versus α and RMSE versus α graphs between these two extreme curves.

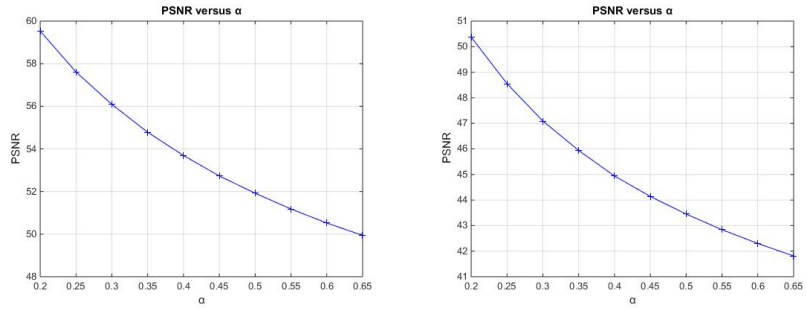


(a) Minimum for "Pepper" image (b) Maximum for "Bear" image

Fig. 17 RMSE vs. α curve.

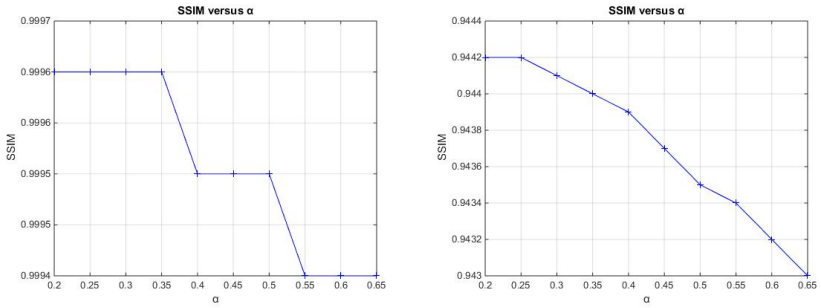
5.3 Extraction Testing

In this subsection, the detector's responses for different watermarked images are shown. For all the watermarked images, 1000 different seed values are tested and the responses to each of these 1000 seed values are tested. If the detector response exceeds the threshold then it means that the detector has detected the presence of a watermark in the image. However, it does not necessarily mean that the image has a watermark. This could be a false positive, which depends on the seed value for which the detector response exceeds that threshold. If the detector response exceeds the threshold at a seed value that is chosen for watermarking in the insertion process then it means that it is a true positive. If the response exceeds the threshold at any other seed value then it means a false positive. It is thus desired to have a sudden peak value (exceeding the threshold) at the desired value and low values at all other seed values. The detector response for minimum PSNR image (for example the "Bear" image) is shown in Figs. (20(a)) and (20(b)) for different values of α.



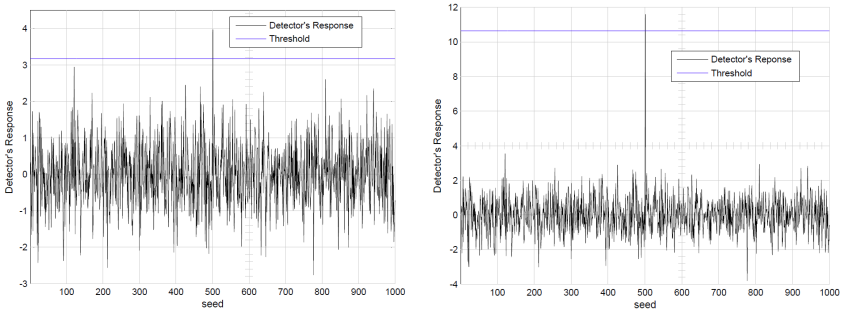
(a) Maximum for "Pepper" image (b) Minimum for "Bear" image

Fig. 18 PSNR vs. α curve.



(a) Maximum for "Pepper" image (b) Minimum for "Bear" image

Fig. 19 SSIM vs. α curve.



(a) ($\alpha = 0.2$) (b) ($\alpha = 0.65$)

Fig. 20 Detector's response for the "Bear" image.

As expected, with higher values of the scaling factor α there is a sharp increase in the detector's response at a particular seed value. In this case, the percentage increase in the detector's response is 192%. This increase comes at the expense of reduced PSNR of the image i.e., there is a decrease of 83.6% in the PSNR value, which implies that the signal quality is reduced. This also illustrates the importance of choosing an optimum value of α to trade off between robustness and detectability of the watermark with image quality. The results suggest that the proposed watermarking algorithm is effective and gives very low false positives and false negatives.

5.4 Estimation of the Embedding Capacity

The estimation of embedding capacity is the process of finding the size of the largest watermark capacity that can be embedded into images. It is one of the most important issues in invisible watermarking. There are potentially many different techniques to estimate the embedding capacity such as multi-pass embedding capacity and Human visual system (HVS). Because the multi-pass embedding capacity requires additional memory [38], HVS is used in this work to quantify the embedding capacity by calculating the perceptual metric in CIELAB [39] to find the color difference between the host images and the watermarked images. From the definition of CIE $L^*a^*b^*$ or CIELAB, L^*_c, a^*_c, b^*_c refers to the host image and L^*_w,a^*_w,b^*_w to the corresponding watermarked images. The total difference of color, which is considered as the tolerance of a color [40], is determined by the following:

$$\Delta total^* = \sqrt{\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2}}, \tag{14}$$

where ΔL^*, Δa^*, and Δb^* are the color differences for each color channel R, G, and B respectively. By separating these channels of the host image O, and transforming them into ΔL^*,Δa^*, and Δb^*, the algorithm separately calculates the color difference of each channel as follows:

$$\Delta L^* = L^*_w - L^*_c \tag{15}$$

$$\Delta a^* = a^*_w - a^*_c \tag{16}$$

$$\Delta b^* = b^*_w - b^*_c \tag{17}$$

From Eqn. 14, the estimation of the watermark embedding capacity will be the total capacity of colors of an image. For brevity, three host images with different sizes were selected and their corresponding watermarked images with value of $\alpha = 0.45$ were used to estimate the embedding capacity. Since the proposed scheme considers just the center quarter of the host image, which is chosen based on the perceptually important region, the focus is in the center quarter of the image to estimate the embedding capacity. Table 5 demonstrates the estimation of embedding capacity. The proposed scheme considers the center quarter of a host image, which provides several advantages in term of robustness and high performance, but at the same time the embedding capacity will be reduced because focusing just in the important region will degrade the image quality and some information will be lost.

5.5 Testing with Different Attacks

This subsection analyzes the detector's response when the watermarked images are subjected to different types of attacks. This reveals how robust the watermarking algorithm is. If

Table 5 Estimation of the embedding capacity

Host Image	Size of Watermarked Image (KB)	Dimensions of Important Region	Embedding Capacity of the Center Quarter (bits)
F16 512 × 512 (37.5KB)	36	128 × 128	432817
Bear 256 × 256 (19.4KB)	11.6	64× 64	309231
Wallpaper 128 × 128 (6KB)	4.22	32× 32	270883

the watermark can be extracted from the attacked, corrupted, and/or modified watermarked image then the watermark is said to have survived, otherwise it has not survived. However, the extracted watermark will be degraded due to such attacks. The watermarked images have been tampered with the build-in functions of the software package ImageMagick®[41]. The attacks that have been tested for this experiment are the following: (1) JPEG compression (2) Sharpen (3) Pixelize, and (4) Noise.

5.5.1 Attack 1: JPEG compression

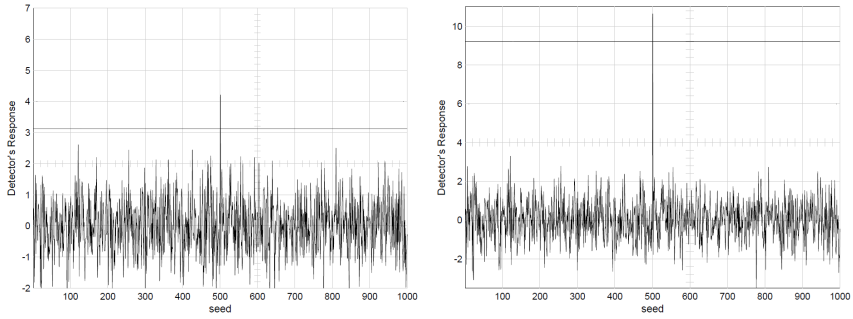
JPEG compression is a widely used algorithm for image compression and is considered an unintentional attack. Various Quality Factors (QF = 75%, 50%, 25% and 10%) are used to compress the watermarked image (shown only for the "Bear" image), and the resulting images are shown in Figs. (21(a)), (21(b)), and (21(c)). The resulting images are then tested for watermark extraction, and Table (6) shows the results. The results show that if higher values of α are used for embedding, then the watermark can resist higher levels of JPEG compression.

(a) QF=10, $\alpha = 0.2$ (b) QF=50, $\alpha = 0.2$ (c) QF=25, $\alpha = 0.65$

Fig. 21 Compressed watermarked image.

Table 6 JPEG Compression result with different QF and α

Tampering Operation	$\alpha = 0.2$	$\alpha = 0.65$
JPEG Compression, QF 10%	Not recognized	Not recognized
JPEG Compression, QF 25%	Not recognized	Survived
JPEG Compression, QF 50%	Survived	Survived



(a) QF=50, $\alpha = 0.2$ (b) Detector's Response (QF=25, $\alpha = 0.65$)

Fig. 22 Detector's response.

5.5.2 Attack 2: Sharpening

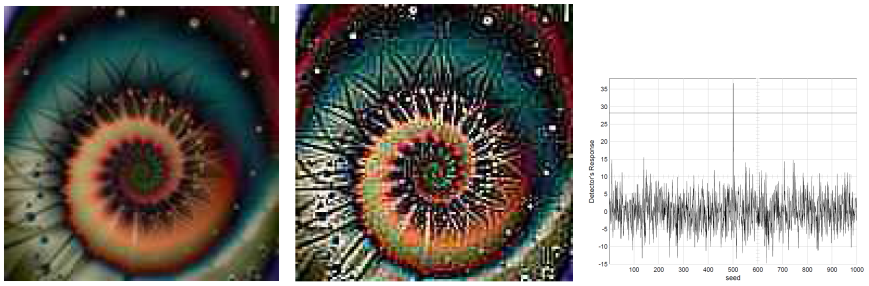
The watermarked images are seen to be robust against the sharpening filter. After applying 100% sharpening then compressing the image by a 25% quality factor, the detector is able to detect the presence of the watermark distinctly. Figs. (23(b)), (23(c)), (24(b)), and (24(c)) show the 100% sharpened images along with the watermarked images and the detector's response.

(a) Watermarked image (b) 100% Sharpened (QF=25) (c) Detector's response (QF=25, 100% sharpening)

Fig. 23 Sharpened watermarked "Bear" image.

5.5.3 Attack 3: Pixelize

The watermarked images are pixelized in the size of 2×2 and 3×3 and then compressed with different QF. Figs. (25(b)), and (25(c)) show the pixelated images along with the watermarked image and the detector's response. The results for pixelization attacks are given in Table (7). The results show that the algorithm is robust with respect to pixelization and larger size pictures are more tolerant to pixelization effects.



(a) Watermarked image (b) 100% Sharpened (QF=25) (c) Detector's response (QF=25, 100% sharpening)

Fig. 24 Sharpened Watermarked "Wallpaper" image.

(a) Wallpaper image (b) 2x2 Pixelate (QF=25) (c) Detector's response (QF=25, 100% sharpening)

Fig. 25 Pixelated watermarked image.

Table 7 Pixelization of different images with different QF.

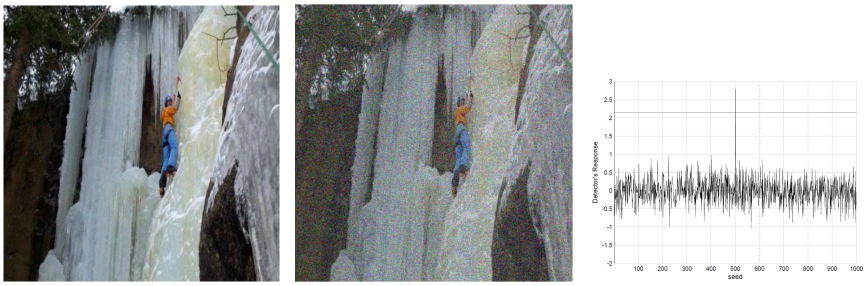
Tampering Operation	**Wallpaper Image** (128×128)	**Bear Image** (256×256)	**Ice Climb Image** (512×512)
2x2 Pixelise, JPEG Compression, QF 25%	Survived	Survived	Survived
3x3 Pixelise, JPEG Compression, QF 90%	Too noisy	Too noisy	Survived

5.5.4 Attack 4: Noise

Gaussian noise is inserted into the image in different amounts and then the image is compressed with different QF. Figs. (26(b)), and (26(c)) show the watermarked image affected by noise and the corresponding detector's response. The effect of noise on the images is given in Table (8). From the table it is inferred that larger size images are more resistant to noise.

5.5.5 Attack 5: Geometric Distortions

Geometric distortion is an important concern in testing the robustness of the watermarking algorithm for copyright protection of digital images. Resizing, rotation, and cropping are considered to be the most common attacks in this class. The robustness of the algorithm was tested against the resize attack. The standard test image Lena of size 512×512 is considered,



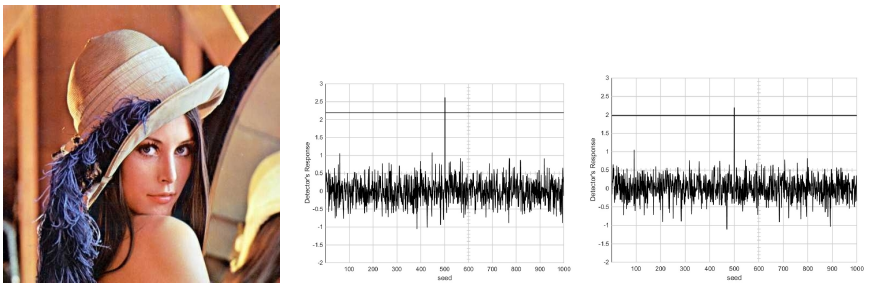
(a) Wallpaper image (b) 40% noise added (QF=25) (c) Detector's response (QF=25, 40% Noise)

Fig. 26 Noise added to watermarked image.

Table 8 Noise on different images with different QF.

Tampering Operation	**Wallpaper Image** (128×128)	**Bear Image** (256×256)	**Ice Climb Image** (512×512)
25% Noise, JPEG Compression, QF 25%	Survived	Survived	Survived
35% Noise, JPEG Compression, QF 90%	Too noisy	Survived	Survived
40% Noise, JPEG Compression, QF 25%	Too noisy	Noisy	Survived

then the watermarked image was resized to a new size of 500×500, 490×490, and 480×480, and compressed with different QF. Figs. (27(a)), (27(b)), and (27(c)) show the resized watermarked images along with the detector's responses. Table (9) shows the results. The results prove that the algorithm is robust with respect to higher vales of α. The main advantage of using the blind algorithm is non-availability of original image information at the detector. After applying the geometric distortions, if the watermark can be extracted it means that a watermark is present, otherwise it means the watermark is not recognized. Frequency domain watermarking is more widely used for copyright protection due to its robustness. The following geometric attacks can be addressed using DCT watermarking using the techniques discussed above. Part of the rows of any watermarked image is mostly removed in cropping. The width of the column can be considered with a width of 8 pixel and the watermarked image can be divided into different groups of information.



(a) Watermarked image, resized to 480×480, JPEG compression (b) Detector's response (α=0.2, QF=25, 480×480 resizing) (c) Detector's response (α=0.65, QF=50, 480×480 resizing)

Fig. 27 Resizing the watermarked "Lena" Image.

Table 9 Resizing Watermarked "Lena" Image with different QF.

Tampering Operation	α	**QF**	**Lena 512X512**	PSNR
Resize to 500×500, JPEG Compression	0.2	25%	Survived	51.23
	0.65	50%	Survived	42.88
Resize to 490×490, JPEG Compression	0.2	25%	Not recognized	49.03
	0.65	50%	Survived	42.31
Resize to 480×480, JPEG Compression	0.2	25%	Not recognized	48.93
	0.65	50%	Survived	42.54

A combination of rotation and cropping attack is used to test the robustness of the algorithm against geometric distortions. Applying different angles in the rotational process then cropping the images, the detector is able to detect the presence of the watermark in images with larger size, which are more resistant to rotation and cropping, as demonstrated in Table (10) and Figs. (28(a)), (28(b)), (28(c),(29(a),(29(b)and (29(c)).

(a) Watermarked image with 90° rotation (b) Watermarked image with 90° rotation, cropping) (c) Detector's response (α=0.65, 90° rotation, cropping)

Fig. 28 Rotation and cropping of the watermarked "Bear" image.

(a) Watermarked image with 30° rotation (b) Watermarked image with 30° rotation, cropping) (c) Detector's response (α=0.65, 30° rotation, cropping)

Fig. 29 Rotation and cropping of the watermarked "Ice Climb" image.

Table 10 Rotation and cropping on different images with different angles.

Tampering Operation	**Wallpaper Image** (128×128)	**Bear Image** (256×256)	**Ice Climb Image** (512×512)
30° rotation, with cropping	Not recognized	Not recognized	Survived
90° rotation, with cropping	Not recognized	Survived	Survived

5.5.6 Attack 6: Collage Attack

Collage attack is considered to be a substitution or counterfeiting attack. It takes advantage of the vulnerability of embedding the watermark independently into a local region of the image with the same mark. An attacker forges a falsified image using parts of a group of images protected by the same authenticator using the same mark. The study in [42] proves the possibility of collage attack even when a logo is not known by using a larger number of images watermarked with the same key and logo to create a new image while preserving their relative spatial location within the image. However the proposed scheme independently embeds the watermark in each block, so the proposed algorithm can effectively resist collage attacks due to the following:

1. Pseudo random noise is used as mark, which is changed for every image. So, no logo can be used to figure out the relative spatial location within a large number of images.
2. The algorithm analyzes five important aspects, as reported in Section 6, for computation of the perceptually important region. Thus, this partial region is selected based on the image and it means that it is different from image to image.

5.6 Comparison with prior works

There is a substantial body of literature available on watermarking. This subsection provides a comparison of the proposed watermarking insertion algorithm to other currently used algorithms. The results ares shown in Table 11. As can be seen from the Table, the proposed algorithm gives higher PSNR when there are no attacks on the watermarked images. From the previous results, it also reveals that the proposed watermark algorithm is robust to most image attacks, including noise.

Table 11 PSNR values for different watermarking algorithm.

S.No.	**Algorithm**	**PSNR**
1	Image adaptive watermarking [28]	40.054 dB
2	Image adaptive watermarking creation [25] Image adaptive watermarking creation [43]	45 dB 35.17 dB
3	Zerotree of wavelet [23]	44.18 dB
4	9/7 biorthogonal wavelet lifting [26]	36.44 dB
5	Robustness of DCT-based watermarking against JPEG compression [44]	34 dB
6	DCT-based watermarking using inter-block coefficient correlation [45]	41.78 dB
7	Metric-based fitness function for robust watermarking [46]	$>$ 41.63 dB
8	**Proposed Algorithm**	$>$ 41.81 dB (in case of code simulation) $>$ 44.37 dB (in case of Simulink® model)

6 Proposed Watermarking Architecture and Simulink®based Simulations

A system-level architecture for the proposed watermarking algorithm is illustrated in Fig. (30). The block shown in the dotted line finds the region where the watermarking needs to be done. Thus, the five important aspects of (1) the center quarter, (2) edginess, (3) texture, (4) contrast, and (5) intensity, are analyzed to find the important region. Indeed, we try to insert the watermark in those areas which are a bit dark. This allows us to hide more information without affecting the quality, and avoid the area where there is drastic change in the color component. The important region varies on an image based on the above factors. This means it is different from one image to another which makes the proposed architecture effectively resist collage attack.

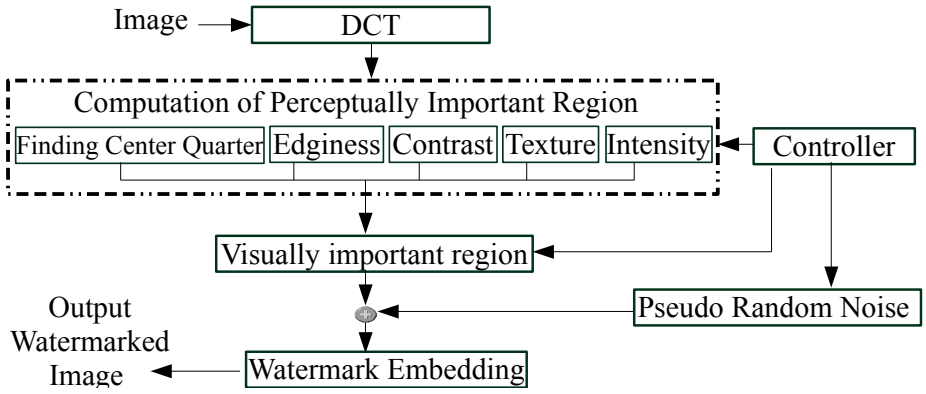


Fig. 30 System Level Architecture of the Proposed Watermarking Algorithm.

6.1 Watermark Insertion Unit

The insertion unit is composed of several sub-modules: a DCT module, a perceptual analyzer, a scale factor, and the insertion module. The DCT module calculates the DCT coefficients of the host image. The controller manages the operation schedules of all other modules and the data flow of the unit. Fig. (31) illustrates the watermark insertion unit.

6.2 Discrete Cosine Transformation (DCT) Unit

The DCT module calculates the DCT coefficients of the image, as shown in Fig. (32). The algorithm splits the image into Y, Cb, and Cr frames, then considers just the Y frame in blocks of size 8×8. A buffer is used to assist in finding the transpose of the 1D row DCT [47].

6.3 MATLAB®/Simulink®Based Modeling

Experimental simulations are performed to perform a comparative analysis of the proposed algorithm with previously published results. The proposed algorithm is implemented as a

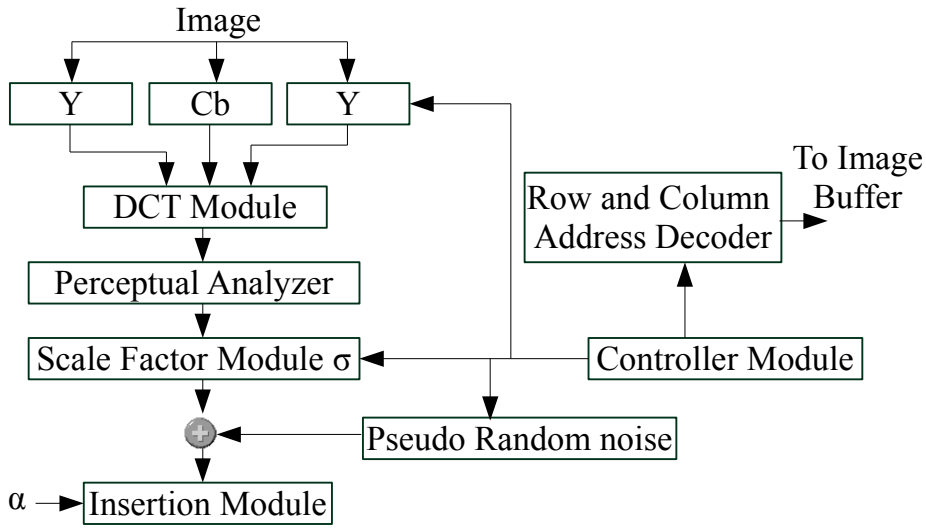


Fig. 31 Watermark Insertion Unit.

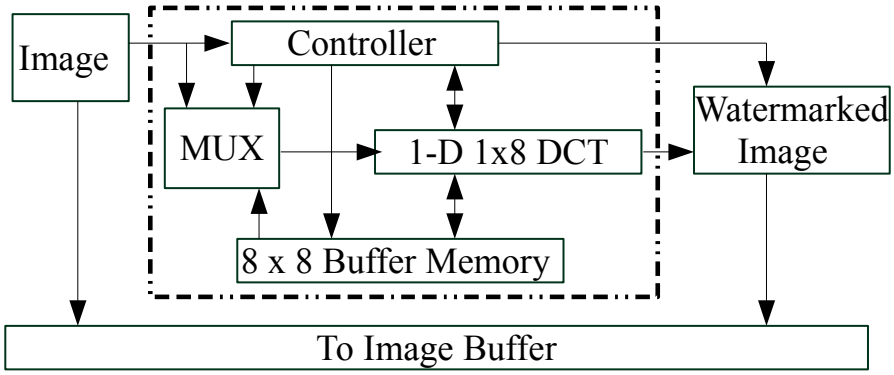


Fig. 32 Discrete Cosine Transformation (DCT) Unit.

fast prototype in MATLAB®/ Simulink® Version 8.3 (R2014a), with the Computer Vision System Toolbox Version 9.7 [47]. The insertion and extraction modules are shown in Fig. 33 and Fig. 34, respectively. The methodology for this high level system modeling is bottom-up. The first step is building function units, then integrating these units into sub-systems, and finally verifying overall system functionality. MATLAB®/Simulink® offers image processing functions and modules that facilitate fast prototyping. Another advantage of using MATLAB®/Simulink® is the availability of function units such as DCT/IDCT, and block processing. In addition, the system-level modeling is accomplished using different modules such as color conversion, and DCT domain compression.

Table 12 and Table 13 show the RMSE and PSNR values for different values of α obtained from Simulink®. From the results it can be seen that the value for PSNR is maintained above 44.37 dB for all cases, which closely matches that of 41.81 dB obtained from

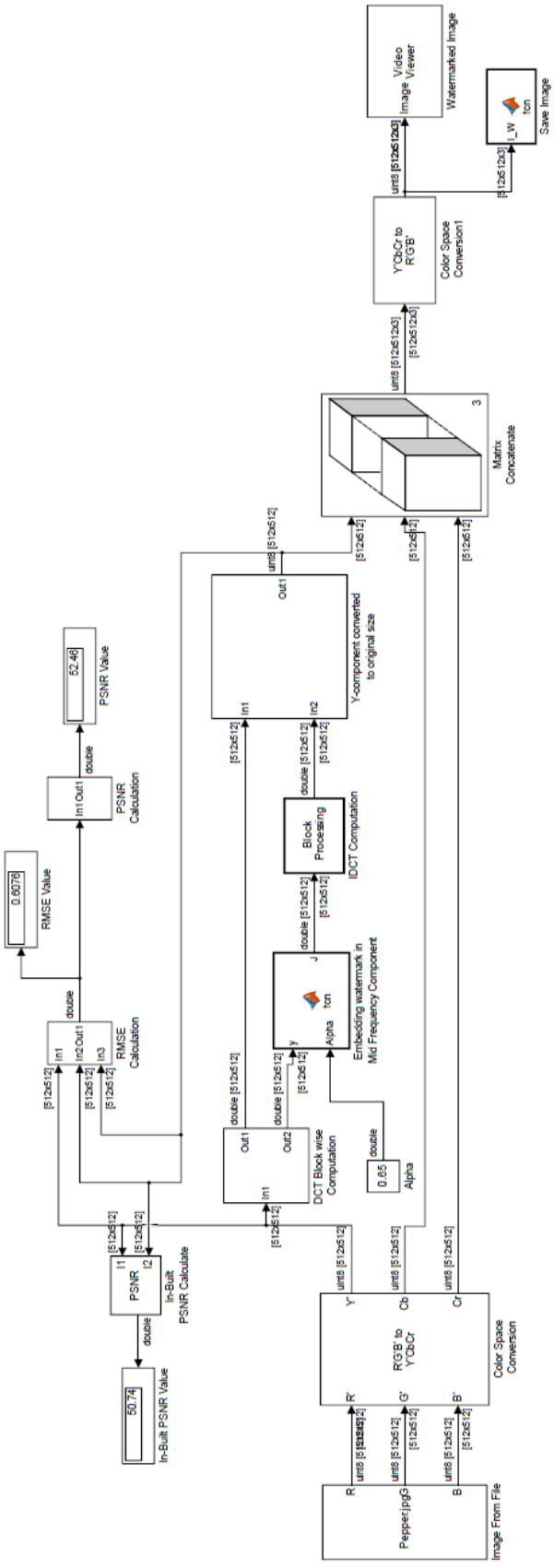


Fig. 33 Watermark insertion algorithm implemented in Simulink®.

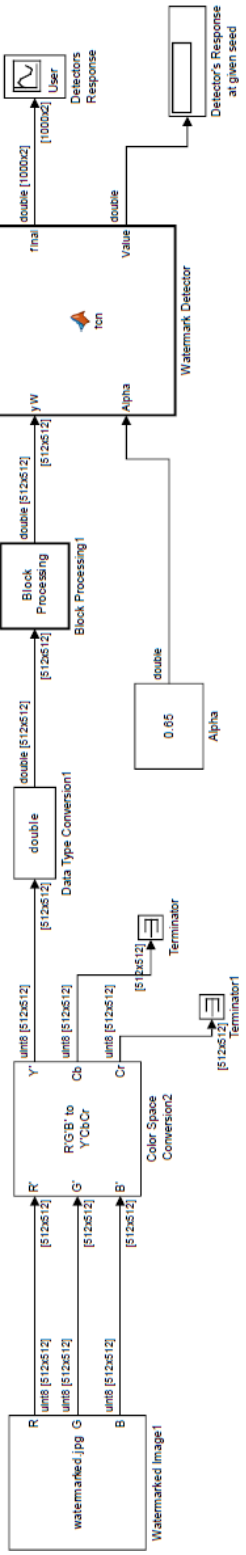


Fig. 34 Watermark detection algorithm implemented in Simulink®.

MATLAB®previously. Similarly, the average PSNR for the "Pepper" image is 54.35 dB, which is the highest of all images.

Table 12 RMSE values of the analyzed images from Simulink®.

α	**0.2**	**0.25**	**0.3**	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**
Lena	0.50	0.56	.62	0.69	0.76	0.83	0.89	0.96	1.03	1.09
Baboon	0.46	0.50	0.55	0.60	0.66	0.71	0.77	0.82	0.88	0.93
Pepper	0.39	0.41	0.43	0.45	0.48	0.50	0.53	0.56	0.58	0.61
F-16	0.43	0.47	0.51	0.56	0.61	0.66	0.71	0.76	0.81	0.86
Forest	0.53	0.61	0.70	0.77	0.85	0.94	1.02	1.10	1.19	1.28
Wallpaper	0.48	0.55	0.60	0.68	0.74	0.80	0.88	0.93	1.01	1.08
Squarrel	0.45	0.49	0.53	0.58	0.63	0.68	0.73	0.79	0.84	0.89
Bear	0.62	0.73	0.84	0.95	1.05	1.16	1.26	1.36	1.45	1.54
Google Map	0.61	0.71	0.81	0.91	1.01	1.10	1.20	1.29	1.38	1.47
Resort	0.54	0.62	0.71	0.79	0.88	0.96	1.05	1.13	1.21	1.28
Ice Climb	0.40	0.43	0.45	0.48	0.51	0.55	0.59	0.62	0.66	0.70

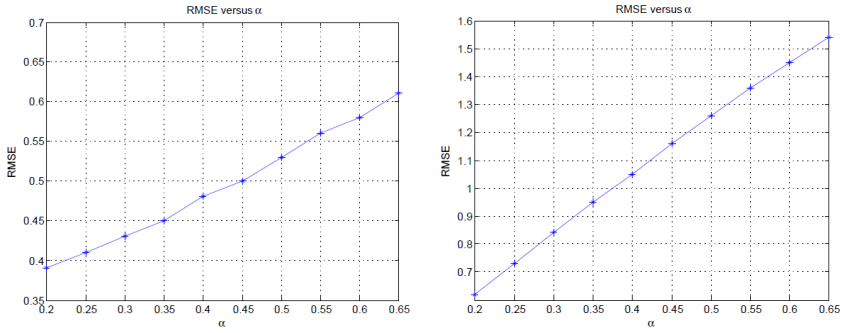
Table 13 PSNR values of the analyzed images from Simulink®.

α	**0.2**	**0.25**	**0.3**	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**
Lena	54.21	53.19	52.22	51.34	50.52	49.79	49.12	48.5	47.91	47.39
Baboon	54.95	54.13	53.29	52.5	51.8	51.09	50.42	49.83	47.91	48.76
Pepper	56.25	55.87	55.46	55.02	54.57	54.11	53.67	53.24	52.85	52.46
F-16	55.42	54.67	53.9	53.15	52.46	51.77	51.13	50.54	49.97	49.44
Forest	53.64	52.46	51.39	50.41	49.54	48.73	47.97	47.28	46.62	46.02
Wallpaper	54.56	53.4	52.52	51.63	50.78	50.02	49.27	48.72	48.09	47.56
Squarrel	55.16	54.39	53.62	52.88	52.13	51.46	50.81	50.23	49.69	49.17
Bear	52.23	50.86	49.66	48.59	47.67	46.85	46.11	45.48	44.89	44.37
Google Map	52.34	51.05	49.96	48.95	48.08	47.28	46.53	45.9	45.34	44.81
Resort	53.45	52.23	51.13	50.15	49.25	48.45	47.73	47.08	46.51	45.97
Ice Climb	56.03	55.55	55.02	54.48	53.9	53.32	52.77	52.25	51.77	51.28

From Tables 12 and 13, it can be seen that the average PSNR value for the "Pepper" image is maximum at 54.35 dB. Similarly, the minimum value for the average PSNR is for the "Bear" image, at 47.67 dB. The Simulink® results match very closely the MATLAB® results, which confirms that the proposed architecture can be used for hardware implementation. From Simulink®results, the graphs of PSNR versus α and RMSE versus α for the "Pepper" and "Bear" images are shown in Figs. 35(a), 35(b), 36(a), and 36(b).

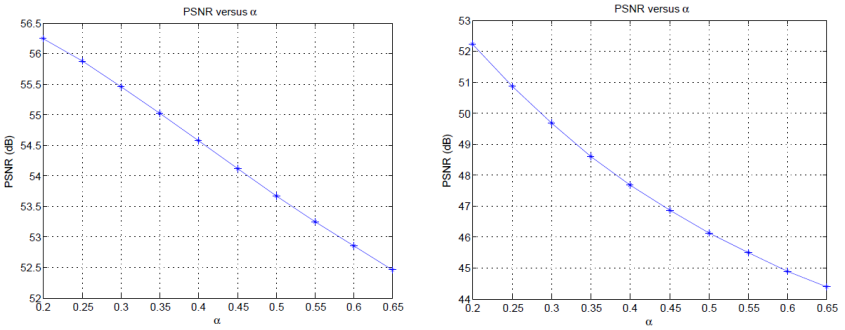
6.4 Comparison of DCT vs DWT

In this subsection, we implement the same system by replacing the DCT in Simulink®with the DWT block and present a comparison of DCT vs. DWT performance of the algorithm. Higher picture quality and higher compression ratios are the main advantages of using the DWT instaed of the DCT [48]. Analysis of images or signals, according to resolution is allowed in wavelet functions. An image in the DWT domain is represented as a sum of wavelet functions, known as wavelets with different scale and location. The data is represented as a



(a) Minimum for Pepper image (b) Maximum for Bear image

Fig. 35 RMSE vs. α curve.



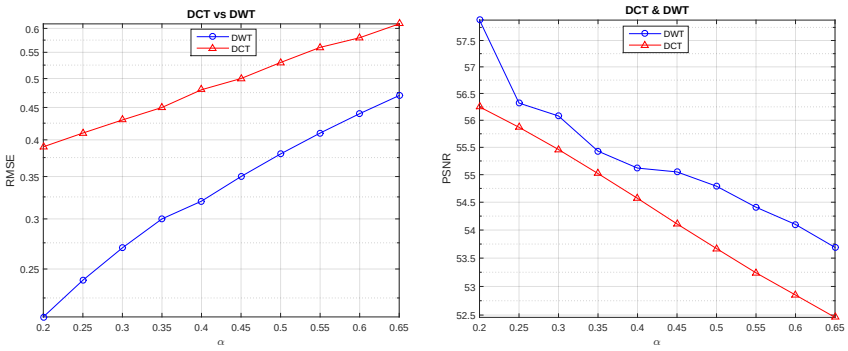
(a) Maximum for Pepper image (b) Minimum for Bear image

Fig. 36 PSNR vs. α curve.

set of low pass (approximate) and high pass (detailed) coefficients [49]. The detailed coefficients are the output of the high pass filter and the approximate coefficients are the output of the low pass filter. This procedure is called one dimensional (1-D) DWT. In the 2-D DWT case, the input is passed through high pass and low pass filters in two directions, both rows and columns. The obtained output is a set of four coefficients HH, LH, HL, LL. The transformation in a column is represented by the second letter and the row transform is represented by the first letter. 'L' represents low pass filter and 'H' represents high pass filter. 'LH' represents the high pass signal at the column and low pass signal at the row. The horizontal elements are present in 'LH'. HH and HL, similarly, contain diagonal and vertical elements [50]. To summarize the procedure of DWT:

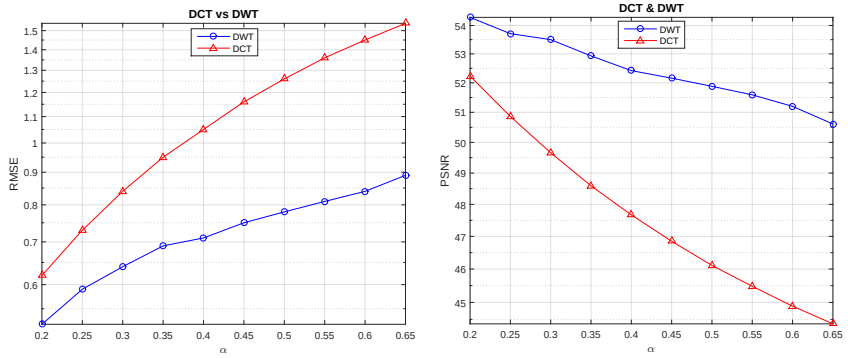
1. Perform low pass filtering and high pass filtering.
2. Obtain the detail and approximation coefficients.
3. Approximate coefficients are separated as HH, HL, LH and LL coefficients.
4. Except for the LL coefficients, all the other coefficients are discarded. Then the 'LL' coefficients are transformed into the second level.
5. Then the coefficients are divided by a constant scaling factor.
6. As a final step, the data is padded with zeros and rescaled and passed through a Wavelet filter.

For brevity only two images, with the highest and lowest values of PSNR and RMSE, are presented here. The results from different transform coding techniques, Discreet Wavelet Transform and Discreet Cosine Transform, are compared in Figs. 37(a), 37(b), 38(a), and 38(b). DWT gives slightly higher quality, however DWT is more complicated than DCT and hardware implementation is much more involved. To reduce this hardware complexity and meet future requirement of low power design, we preferred to use DCT.



(a) RMSE vs. α curve (b) PSNR vs. α curve

Fig. 37 Pepper image.



(a) RMSE vs. α curve (b) PSNR vs. α curve.

Fig. 38 Bear image.

7 Conclusions and Directions for Future Research

An effective invisible robust blind watermarking technique for copyright protection has been presented in this paper where the watermarking is inserted in color images in their luminance (Y) component. In order to make the watermarking algorithm robust to attacks, the

watermark is inserted in the mid frequencies. Analyzing the results, it is clear that the proposed watermarking algorithm is invisible for the obtained PSNR values. Comparing with other watermarking schemes, the PSNR values obtained under no attack conditions in the proposed algorithm are much higher than the PSNRs obtained by image adaptive watermarking [28] (40.054 dB), image adaptive watermark creation [43] (35.17 dB), zerotree of wavelet [23] (44.18 dB), and 9/7 biorthogonal wavelet lifting [26] (36.44 dB). Exhaustive tests conducted on nine different images prove the image quality and recognizability of the extracted watermark under various types of attacks such as JPEG compression, sharpening, pixelization and noise insertion. Furthermore, the results show that the effectiveness of the proposed watermarking algorithm increases with the size of the host image. For future work, power efficient versions of the algorithm along with a real-time implementation are under progress. A VLSI architecture and chip for the proposed invisible robust blind watermarking technique will be the next topic of investigation.

8 Acknowledgments

The authors would like to thank University of North Texas graduates Sahasan Naraharisetti and Shital Joshi for their help in this research.

References

1. S. P. Mohanty, ISWAR: An Imaging System with Watermarking and Attack Resilience, CoRR abs/1205.4489.
URL http://arxiv.org/abs/1205.4489
2. S. Naraharisetti, Region Aware Dct Domain Invisibile Robust Blind Watermarking For Color Images, Master's thesis, Dept. of Computer Science and Engineering, University of North Texas, Denton, TX 76203 (December 2008).
3. S. P. Mohanty, Watermarking of Digital Images, Master's thesis, Dept. of Electrical Engineering, Indian Institute of Science, Bangalore-560012 (1999).
4. P. H. W. Wong, O. C. Au, Y. M. Yeung, Novel Blind Multiple Watermarking Technique For Images, IEEE Transactions on Circuits and Systems for Video Technology 13 (8) (2003) 813–830. doi:10.1109/TCSVT.2003.815948.
5. T. Zong, Y. Xiang, S. Guo, Y. Rong, Rank-Based Image Watermarking Method With High Embedding Capacity and Robustness, IEEE Access 4 (2016) 1689–1699.
6. B. L. Gunjal, S. N.Mali, Comparative performance analysis of digital image watermarking scheme in DWT and DWT-FWHT-SVD domains, in: Proceedings Annual IEEE India Conference (INDICON), 2014, pp. 1–6.
7. A. K. Hawlader, Moniruzzaman, F. Hossain, A novel robust blind digital watermarking scheme based on blocking probability, in: Proceedings International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), 2014, pp. 1–6.
8. S. Hamad, A. Khalifa, Block based robust blind image watermarking using discrete wavelet transform, in: Proceedings IEEE 10th International Colloquium on Signal Processing & its Applications (CSPA), 2014, pp. 58 – 61.
9. N.-T. Le, T. Le, Y. M. Jang, Optical camera communications based invisible watermarking technique, in: Proceedings International Conference on Information Networking (ICOIN), 2016, pp. 479 – 481.
10. D. O. Munoz-Ramirez, R. Reyes-Reyes, V. Ponomaryov, C. Cruz-Ramos, Invisible Digital Color Watermarking Technique In Anaglyph 3D Images, in: Proceedings 12th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), 2015, pp. 1–6.
11. S. Ghosh, S. C. S. P. Maity, H. Rahaman, A new algorithm on wavelet based robust invisible digital image watermarking for multimedia security, in: Proceedings International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV), 2015, pp. 72 – 77.

12. S. Kaur, M. Lal, An invisible watermarking scheme based on Modified Fast Haar Wavelet Transform and RSGWPT, in: Proceedings 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS), 2015, pp. 1–5.
13. Pushpalatha, S. Shravan, FPGA implementation of an efficient 2D 5/3 Lift DWT based invisible watermarking technique, in: Proceedings Annual IEEE India Conference (INDICON), 2016, pp. 1–6.
14. I. K. Jelena Music, E. Franca, Wavelet based watermarking approach in the compressive sensing scenario, in: Proceedings 4th Mediterranean Conference on Embedded Computing (MECO), 2015, pp. 315 – 318.
15. Y. Pathak, S. Dehariya, A more secure transmission of medical images by two label DWT and SVD based watermarking technique, in: Proceddings International Conference on Advances in Engineering and Technology Research (ICAETR), 2014, pp. 1–5.
16. N. Bansal, A. Bansal, V. K. Deolia, P. Pathak, Comparative analysis of LSB, DCT and DWT for Digital Watermarking, in: Proceddings 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015, pp. 40–45.
17. M. L. Miller, G. J. Doerr, I. J. Cox, Applying Informed Coding and Embedding To Design A Robust High-capacity Watermark, IEEE Transactions on Image Processing 13 (6) (2004) 792–807. doi:10.1109/TIP.2003.821551.
18. X.-Y. Liu, G. Kun, W.-F. Chen, A Blind Watermarking Optimal Detection Based on the Wavelet Transform Domain, in: Proceedings of the International Conference on Machine Learning and Cybernetics, 2007, pp. 1779–1783. doi:10.1109/ICMLC.2007.4370436.
19. Z. Guannan, W. Shuxun, W. Quan, An Adaptive Block-Based Blind Watermarking Algorithm, in: Proceedings of the 7th International Conference on Signal Processing, 2004.
20. Z. Erhu, Z. Fan, Adaptive Image Blind Watermarking Method Based on Zerotree of Wavelet, in: Proceedings of the 8th International Conference on Electronic Measurement and Instruments, 2007, pp. 2–799–2–802. doi:10.1109/ICEMI.2007.4350801.
21. P. Yu, Blind Watermarking Scheme Based on the Sign of Wavelet Coefficients, in: Proceedings of the 8th International Conference on Signal Processing, 2006. doi:10.1109/ICOSP.2006.345583.
22. B. C. Choi, D. I. Seo, A Statistical Approach For Optimal Watermark Coefficients Extraction in HVS-Based Blind Watermarking System, in: Proceedings of the 7th International Conference on Advanced Communication Technology, 2005, pp. 1085–1088. doi:10.1109/ICACT.2005.246147.
23. L. Zhi-Bo, F. Jiu-Lun, Z. Hong-Cai, A Blind Watermarking Algorithm Based On Wavelet Lifting Transform, in: Proceedings of the 7th International Conference on Signal Processing, 2004, pp. 843–847. doi:10.1109/ICOSP.2004.1452795.
24. J.-C. Yen, H.-C. Chen, J.-H. Juan, Blind Watermarking Based on the Wavelet Transform, in: Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, 2006, pp. 474–478. doi:10.1109/PDCAT.2006.40.
25. X.-H. Qiao, S.-X. Wang, Q. Wen, Z. Xu, A Robust Watermarking Algorithm Adopting Double Embedding, in: Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2006, pp. 63–66. doi:10.1109/IIH-MSP.2006.265120.
26. A. Khalfallah, F. Kammoun, M. Bouhlel, C. Olivier, A New Scheme of Watermarking in Multi-resolution Filed By 5/3 Wavelet: Family Signature Combined With The Adapted Embedding Strength, in: Proceedings of the 2nd International Conference on Information and Communication Technologies, 2006, pp. 1145–1152. doi:10.1109/ICTTA.2006.1684536.
27. L. Yongliang, X. Yang, H. Yao, T. Huang, W. Gao, Watermark Detection Schemes With High Security, in: Proceedings of the International Conference on Information Technology: Coding and Computing, 2005, pp. 113–117. doi:10.1109/ITCC.2005.295.
28. R. Safabakhsh, S. Zaboli, A. Tabibiazar, Digital Watermarking on Still Images Using Wavelet Transform, in: Proceedings of the International Conference on Information Technology: Coding and Computing, 2004, pp. 671–675. doi:10.1109/ITCC.2004.1286543.
29. E. Ganic, A. M. Eskicioglu, Robust DWT-SVD Domain Image Watermarking: Embedding Data in All Frequencies, in: Proceedings of the Workshop on Multimedia and Security, 2004, pp. 166–174.
30. A. Piper, R. Safavi-Naini, A. Mertins, Resolution and Quality Scalable Spread Spectrum Image Watermarking, in: Proceedings of the 7th Workshop on Multimedia and Security, 2005, pp. 79–90.

31. N. Terzija, W. Geisselhardt, Digital Image Watermarking Using Complex Wavelet Transform, in: Proceedings of the Workshop on Multimedia and Security, 2004, pp. 193–198.
32. S. Zaboli, M. S. Moin, CEW: A Non-Blind Adaptive Image Watermarking Approach Based on Entropy in Contourlet Domain, in: Proceedings of the IEEE International Symposium on Industrial Electronics, 2007, pp. 1687–1692. doi:10.1109/ISIE.2007.4374858.
33. G. Lavoue, F. Denis, F. Dupont, Subdivision surface watermarking, Computers & Graphics 31 (3) (2007) 480–492.
34. S. P. Mohanty, N. Pati, E. Kougianos, A Watermarking Co-Processor for New Generation Graphics Processing Units, in: Digest of Technical Papers International Conference on Consumer Electronics, 2007, pp. 1–2. doi:10.1109/ICCE.2007.341552.
35. G. P. Betancourth, A. Haggag, M. Ghoneim, T. Yahagi, J. Lu, Robust Watermarking in The DCT Domain Using Dual detection, in: Proceedings of the IEEE International Symposium on Industrial Electronics, 2006, pp. 579–584. doi:10.1109/ISIE.2006.295523.
36. M. Barni, F. Bartolini, V. Cappellini, A. Piva, A DCT-domain system for robust image watermarking, Single Processing Journal 66 (1998) 357–372.
37. Z. Wang, B. A. C, S. H. R, S. E. P, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (2004) 1057–7149.
38. R. Iyer, R. Borse, S. Chaudhuri, Embedding capacity estimation of reversible watewatermark schemes, Indian Academy of Science 39 (2014) 1357–1385.
39. P. P. (E-1.3.1), Colorimetry, 2nd edition, Central Bureau of the CIE, AustriaCIE Publication 15.2 (1986).
40. N. Dharwadkar, B. Amberker, Estimating the embedding capacity of a color image using Color Difference, in: Seventh International Conference on Wireless And Optical Communications Networks (WOCN), 2010, pp. 1 – 5. doi:10.1109/WOCN.2010.5587327.
41. ImageMagick®, http://www.imagemagick.org.
42. J. Fridrich, M. Goljan, N. Memon, Cryptanalysis of the Yeung-Mintzer fragile watermarking technique, JOURNAL OF ELECTRONIC IMAGING 11 (2002) 262–274.
43. S. P. Mohanty, B. K. Bhargava, Invisible Watermarking Based on Creation and Robust Insertion-Extraction of Image Adaptive Watermarks, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 5 (2). doi:10.1145/1413862.1413865.
44. S. D. Lin, S.-C. Shie, J. Guo, Improving the robustness of DCT-based image watermarking against JPEG compression, ELSEVIER Computer Standards & Interfaces 32 (1-2) (2010) 54–60.
45. C. Das, S. Panigrahi, V. K. Sharma, K. Mahapatra, A novel blind robust image watermarking in DCT domain using inter-block coefficient correlation, ELSEVIER International Journal of Electronics and Communications 68 (3) (2014) 244–253.
46. A. M. Abdelhakim, H. I. Saleh, A. M. Nassar, Quality metric-based fitness function for robust watermarking optimisation with Bees algorithm, IET Image Processing 10 (3) (2016) 247 – 252.
47. S. P. Mohanty, E. Kougianos, Real-Time Perceptual Watermarking Architectures for Video Broadcasting, Journal of Systems and Software 84 (5) (2011) 724–738. doi:10.1016/j.jss.2010.12.012.
48. R. C. Gonzalez, R. E. Woods, S. L. Eddins, Digital Image Processing UsingMATLAB, Prentice Hall, 2004.
49. A. Kaur, J. Kaur, Comparision of Dct and Dwt of Image Compression Techniques, International Journal of Engineering Research and Development 1 (4) (2012) 49–52.
50. L. Chen, C. Huang, C. Chen, C. Cheng, VLSI Design of Wavelet Transform: Analysis, Architecture and Design Examples, College press, 2007.