Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

An imperceptible spatial domain color image watermarking scheme

Jobin Abraham *, Varghese Paul

Mahatma Gandhi University, Kottayam, India

ARTICLE INFO

Article history:
Received 19 May 2016
Revised 9 December 2016
Accepted 14 December 2016
Available online 23 December 2016

Keywords:
Color image watermarking
Spatial domain
Watermark
Embedding
Extraction
Attacks

ABSTRACT

The paper proposes a novel scheme for color image watermarking. Spatial domain techniques are used here for embedding the watermark information to generate high quality watermarked image. Spatial domain methods are popular with fragile watermarking techniques that mostly use two or three least significant image bits for storing the recovery information. Here spatial domain methods are further explored to develop a robust mechanism for copyright protection. The method presented gradually spreads the watermark information over a region of pixels as implemented by the transform domain techniques. Thus the method is designed to deliver the two essential features required of watermarking systems namely, high image quality and high robustness to attacks. Also after watermark embedding it is ensured that the alterations in one color component are well compensated and no color difference or variations is visually evident. For watermark embedding and color compensation two masks are proposed. The algorithm is experimentally analyzed using various quality metrics and watermark removal attacks. The results prove the model support imperceptible watermarking and high resilience to attacks.

1. Introduction

The wide use of Internet and digital data transfer has lead to several protection issues pertaining to the ownership of published digital image and other digital resources. Availability of advanced image processing tools has made it much easier to download and edit the digital images. In this present context digital watermarking has received increased attention as an effective tool for protecting ones copyrights and ownerships.

In digital watermarking the digital contents are embedded with a unique identification signal known as watermark which can be extracted later and used for authentication (Shiguo et al., 2009). The host digital contents can be any multimedia such as audio, image, video or text. And whenever watermarked images are found to be illegally reused the embedded watermark can be extracted to ascertain the ownership claims.

The three prime applications of image watermarking is copyright protection, authentication and tamper recovery. In copyright protection an identification mark of the resource owner is embedded in the digital image. The hidden identification information can be retrieved anytime later to prove the ownership rights. The methods in Bedi et al. (2013) and Himeur et al. (2012) presents schemes for the copyright protection of the digital video and audio resources. Authentication of image contents using watermarking techniques is also possible. Some features of the image are extracted and are embedded in the image itself. And when suspicions arise a verification of image feature versus the embedded information can authenticate the contents against malicious modifications. Another important application is tamper recovery where the illegally modified contents in the image are recovered to the original form. Most of the published works deal with authentication and tamper recovery as two stages in watermarking process. The contents are authenticated in the first stage and if tampering is detected those portions are recovered in the second stage of detection process.

There are different types of watermarking schemes. One type is robust watermarking where the watermark is strongly embedded to remain resilient to attacks. Hence the robust watermarking techniques are mostly used for copyright protection. The second type is the fragile watermarking schemes used for authentication and tamper detection (Vidhyasagar et al., 2005). When the images suffer unauthorized tampering it is possible to identify those regions and subsequently the damaged regions can be recovered using the hidden recovery watermark (Lin et al., 2005). Technically watermarking schemes can be classified into spatial domain

* Corresponding author.
E-mail addresses: jnabpc@gmail.com (J. Abraham), vp.itcusat@gmail.com (V. Paul).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2016.12.004
1319-1578/

methods and transforms domain methods. Spatial domain methods operate at pixel level (Zhicheng et al., 2006) and transform domain methods uses a mathematical tool as Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT) to convert the pixel values to a set of correlated values and watermark embedding operation is carried out thus leaving a deeper impact over a region of values within the image. Transform based grayscale image watermarking techniques using DCT, DFT or DWT are presented in (Parthasarathy and Kak, 2007; Keqiang and Huihuam, 2011; Manikanda and Ayyasamy, 2014; Chun-Hung et al., 2014; Chih-Chin and Cheng-Chih, 2010; Quing and Jun, 2012). There are methods that use a combination of LSB replacement method and transform methods. The methods in Mei et al. (2015) and Jungpeng et al. (2013) use LSB techniques to embed the watermark but employs a preprocessing stage using one of the transform tool to extract some image features for generating the watermark information.

A spatial domain method loosely based on the principles of DCT is presented in Quingtang et al. (2013a). The color image watermarking scheme transform the image into YC_rC_b color space and directly adds an estimated value to all pixels in the blocks in Y channel in the spatial domain. Another spatial domain method that replaces two to three LSB bits with recovery and authentication information is discussed in Liu (2012).

A color image watermarking mechanism using two-level DCT is proposed in Quingtang et al. (2013b). Here a color image is first transformed into YIQ model and the luminance component Y is used to embed a 32 × 32 sized watermark. Two-level DCT is applied to all non-overlapping 8 × 8 blocks in Y and embeds the watermark in the transformed coefficients. The method however delivers watermarked images with slightly lower PSNR value.

The method in Razieh and Ali (2016) uses a watermark generated from a selected region of interest (ROI) in the host image. Approximate DWT coefficients from ROI are chosen and embedded in the host image itself. The low level frequency coefficients from DWT operation on 8 × 8 blocks in the image is utilized for enforcing watermark embedding. The coefficient with minimal difference with the coefficient to be embedded is considered for embedding. However this makes it essential that the block and the coefficient number used for embedding should be stored as a key and must be made available at the time of extraction.

The method presented in Huynh-The et al. (2016) describes a color image watermarking scheme that uses DWT for embedding watermark signal. Four-level DWT is applied to each host image red, green and blue components and the coefficients in LH and HL band are utilized to embed the watermark information. The difference between LH and HL coefficients is computed for each channel and coefficients with the smallest difference are used to encode 0 and those with largest difference are used for embedding 1-bits.

Though recently there are some works on color image watermarking the majority of the published works use grayscale images as host images for embedding the watermark information. Some works suggest that color images can be treated as grayscale image watermarking and each color component has to be processed in the same fashion separately (Sajjad et al., 2014). However in Internet the use of color images are more common and hence specific issues in watermarking color images needed to be addressed. Since color images contain more information than the grayscale counterparts, color image embedding has different scale of impact. This paper aims to develop a watermark embedding scheme that handles color images more efficiently taking care of individual color channels sensitivity to human eyes. The following of the paper is organized as follows. The section 2 presents a novel method for watermark embedding in color image and section 3 presents experimental analysis of the proposed algorithm. And finally, section 4 concludes the paper.

2. The method

A method for digital image watermarking is discussed. The objective of the paper is to devise a scheme that can watermark color image without significantly degrading the quality of the image and changing the perceptual color. Also all image blocks are embedded with watermark since in certain applications like tamper recovery it is important that all blocks must be watermarked to make the authentication or recovery possible.

2.1. Selection of embedding region

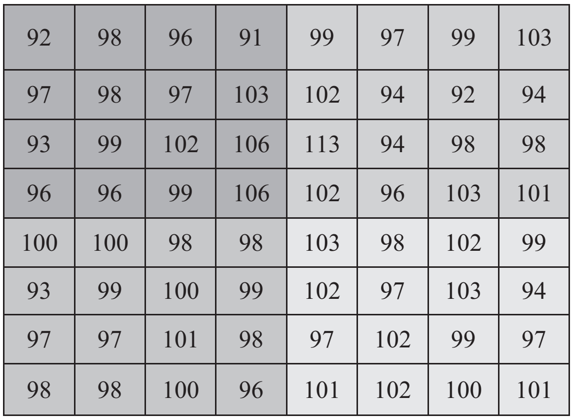
The watermark signal is equivalent to an external noise signal and hence the embedding of watermark must be done in such a way that the distortions introduced are oblivious to human eyes. In most discussed works watermark signal is hidden in certain selected blocks carefully selected from the given image. However, in certain applications this block selection may not be pragmatic as it is essential that the watermark signal has to be embedded to each and every block in the host image. Hence ideal regions in the block must be determined for safely integrating the watermark bit. The proposed algorithm uses a method referred to as SIRD, Simple Image Region Detector, for estimating the most appropriate region within an image block rather than discarding the entire block.

Fig. 1 shows an image block of size 8 × 8. The block is further decomposed into sub-blocks of size 4 × 4, which are denoted as regions R_1, R_2, R_3 and R_4 starting from the top left end. Next depending upon the sub-regional characteristics one region R_s is identified that can better yield to watermark embedding compared to the peer regions in the block. Algorithm 1 explains the procedure for R_s selection. The algorithm is time efficient as only four computations per region are required. Hence for trivial image processing applications where time can be a constraint the proposed algorithm is effective. Though the algorithm is seen to using a diagonal elements test, the block search results can be improved by adopting a two-way diagonal test. In the above example a block from Lena image at coordinate positions 81–88 is shown. And after region analysis, region R_1 is identified to be the most appropriate compared to the remaining segments in the block.

```
Algorithm 1: SIRD
Input: Image block sized mxn, BQ
Decompose BQ into four non-overlapping segments, Rs, s =
  {1, 2, 3, 4}
Set i = 1, j = 1, z1 = m/2, z2 = n/2
for s = 1
  $SD_s = 0$
    for t = i to z1
    for u = j to z2
        if (t= =u), $SD_s = SD_s + | R_s(t, u) − R_s(t+1,u+1)|$
    end u
    end t
end for s
Find x = s of max{SDs}
Output: region Rx
```

2.2. Embedding masks

The color image watermark embedding algorithm uses two masks M_1 and M_2. The masks are used to distribute the watermark information to the neighboring pixels in the selected region. In most schemes, the LSB of the selected image pixel alone gets mod-



92	98	96	91	99	97	99	103
97	98	97	103	102	94	92	94
93	99	102	106	113	94	98	98
96	96	99	106	102	96	103	101
100	100	98	98	103	98	102	99
93	99	100	99	102	97	103	94
97	97	101	98	97	102	99	97
98	98	100	96	101	102	100	101

Figure 1. Local region selected within the block.

ified which may transform them to an intensity level much different from the neighboring members. This modification may make the embedded pixel uncorrelated and can lead to the effects akin to salt and pepper noise addition. Hence to compensate for such degrading effect embedding masks are suggested. The mask distributes the intended alteration to N-8 neighbors thereby gradually distributing the net variation. This ensures that the impacts of alterations are not noticeable and remain insensitive to human eyes.

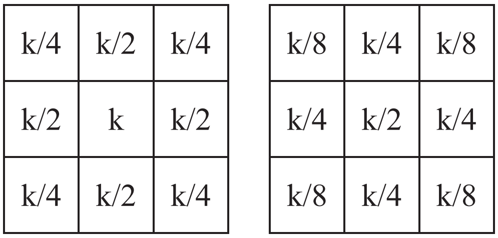
Fig. 2 shows the two masks used while embedding. Mask M_1 is used with the embedding channel and mask M_2 is used for modifying the other color channels to compensate for the variations introduced in the embedding channel. Blue color component has greater insensitivity to human eyes compared to R and G components. Hence B component is used as the embedding channel. Mask M_1 is applied on B channel during the watermark integration and M_2 is applied with the other two components considering the same watermark bit. The combined use embeds the watermark as well as compensates for the changes in non-embedding channels. This ensures the color information in the actual image is effectively retained after the watermark bit embedding process.

$$M_x(i,j) = \mathrm{ceil}(M_x(i,j)) \quad (1)$$

In the above masks, k is the strength factor. The masks can be adjusted to deliver higher embedding strength using a higher k value. And to avoid possible decimal value ceil function in (1) is used to round the values to the nearest largest integer. It may also be noted that the values in M_2 is less than M_1. This ensures that maximum variations are distributed to embedding B component over the other two sensitive channels.

2.3. Watermark preprocessing

Watermark signal is preprocessed to take advantage of redundancy and to increase the watermark detection probability. The watermark is decomposed into four sub-images. Each sub-image is intended to be embedded into each quadrant region in the host image. Assuming the watermark size is PxQ, once decomposed sub-image size reduces to P/2XQ/2. Assuming that the sub-image watermarks are W_1, W_2, W_3 and W_4; each W_i is embedded into the part Q_i of the host image during the following stage of watermark embedding. The equations below (2) are used to generate sub-images, W_i.



k/4	k/2	k/4
k/2	k	k/2
k/4	k/2	k/4

a)

k/8	k/4	k/8
k/4	k/2	k/4
k/8	k/4	k/8

b)

Figure 2. (a) Mask1 (M_1) (b) Mask2 (M_2).

$$\begin{aligned} W_1 &= W(1+m, 1+n) \\ W_2 &= W(1+m, 2+n) \\ W_3 &= W(2+m, 1+n) \\ W_4 &= W(2+m, 2+n) \end{aligned} \quad (2)$$

where m = 0, 2, 4...(P/2–2) and n = 0, 2, 4...(Q/2–2).

2.4. The proposed scheme

The proposed algorithm is meant for watermark signal embedding in a color image. The watermarking process comprises two stages, watermarking embedding and watermark extraction. The block diagrams for the two processes are shown in Figs. 3 and 4. The method uses the RGB color space to perform the watermark process.

The embedding process integrates the watermark logo image in all regions in the cover image. For this, a block based approach is used and each block is subjected to a sub-region selection process using SIRD prior to watermark bit embedding. The pixels in the identified minor region are then modified to represent the watermark bit. Two embedding masks M_1 and M_2 are used while watermark embedding. Mask M_1 modulates blue component to the tune of the watermark bit and M_2 is the compensating mask that adjusts red and green color components.

Watermark extraction is non-blind and the original image is used by the extraction algorithm to retrieve the hidden watermark signal. Once the exact region in which the embedding was introduced is identified the watermark bit is read out for ascertaining the ownership rights. Subsequent readings from all image blocks will deliver four set of watermarks. The individual watermarks from each quarter of the image are in fact the watermark sub-images. Hence the extracted four watermarks are reshaped to form the final output watermark.

2.4.1. Watermark embedding algorithm

A color image of size MxN is embedded with a binary watermark logo of size PxQ. The detailed steps for watermark embedding are outlined below.

Step 1. Input the image, I and the watermark signal W.

The color image uses 24 bits per pixel to represent the color content. The three channels R, G and B comprise 8 bit each. The Blue component is employed as the embedding channel for hiding the watermark information as blue component is more insensitive to human eyes. The watermark signal is a binary logo image of size P = M/8 and Q = N/8.

Step 2. Preprocess the watermark and convert into four one dimensional array, W_i, $i = \{1,...4\}$.

The watermark signal is decomposed into four sub-images by assigning pixels in a 2 × 2 block to separate sub-images. Hence the size of each sub-image watermark is (P/2)X(Q/2). The resulting watermark sub-images W_i is then transformed to the form of one dimensional array so that bits can be easily passed to the procedure that handles bit embedding. The watermark string length is q = 1, 2....(P/2) * (Q/2).

Step 3. Divide I into four quarters each of size M/2xN/2, Q_j, j = {1,...,4}.

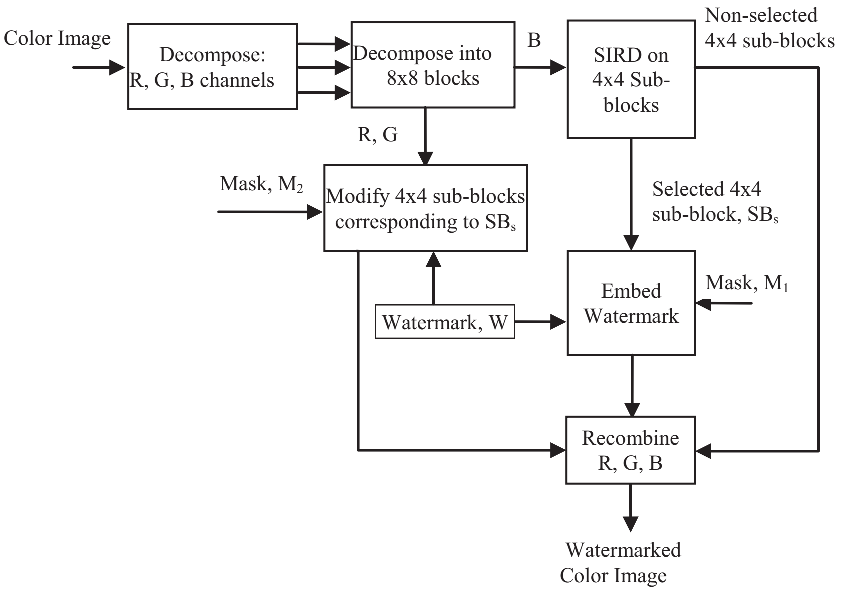

Color Image
Decompose: R, G, B channels
Decompose into 8x8 blocks
B
SIRD on 4x4 Sub-blocks
Non-selected 4x4 sub-blocks
R, G
Mask, M₂
Modify 4x4 sub-blocks corresponding to SBs
Selected 4x4 sub-block, SBs
Embed Watermark
Mask, M₁
Watermark, W
Recombine R, G, B
Watermarked Color Image

Figure 3. Embedding process.

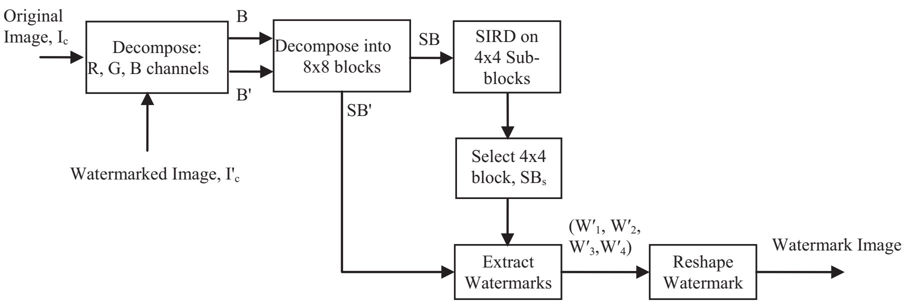


Figure 4. Extraction process.

The image is divided into four non-overlapping quadrant regions known as Q1 to Q4 from left top corner in clock-wise direction. Next, each quarter region is embedded with each copy of the watermark sub-image.

Step 4. Decompose Q_j into R, G and B components, Q_{Rj}, Q_{Gj} and Q_{Bj}.

The quarter region under consideration is decomposed into three constituent color channels. Each value now represents variation in intensity along a single color dimension.

Step 5. Select Q_{Bj} and embed the part of the image Q_{Bj} with W_i, i==j.

The B channel contributes to the embedding blocks whose pixels are modulated in accordance with the watermark information. As there are four quarter regions, each region is embedded with one of the four watermark sub-image. Hence, the first quarter region Q_{B1} will be embedded with W_1 and so on.

Step 6. Decompose each quarter Q_{Bj} into 8×8 blocks, BQ_{BjK}.

The quarter regions are decomposed into non-overlapping 8×8 blocks. The number of possible blocks within one Q_{Bj} is $k = M/16 * N/16$. Also, decompose Q_{Rj} and Q_{Gj} into 8×8 blocks. Hence under the region Q_j there are 3 k blocks in total considering all the three components Q_{Rj}, Q_{Gj} and Q_{Bj}.

Step 7. Estimate most suitable area R_s in the block BQ_{Bjk} using Algorithm 1.

The algorithm looks for a region with maximum variation within the block scanned. The output from this step is 4×4 sub-regions (R_{sB}) identified from B channel. Let the two segments at identical positions taken from the two composite channels, R and G, be R_{sR} and $R_{sG.}$

Step 8. Embed the watermark using the embedding algorithm.

Algorithm 2 details the steps for embedding the watermark, W_{iq}. For embedding bit 1 the values defined in mask M_1 is added and to embed 0 the values in M_1 is discounted from the actual values. Also, the corresponding segments under R_{sR} and R_{sG} are modified using the compensation mask M_2.

Step 9. Continue from step 7 from $k = k + 1$ for embedding the subsequent watermark bit W_{iq}.

All blocks under Q_j is marked by repeating the above two steps 7 and 8. And when $k{=}{=}M/16 * N/16$ all blocks are embedded and proceed for the next image portion.

Step 10. Continue from step5 for the next quarter region Q_j with $i = i + 1$ and $j = j + 1$ until $i = j = 4$.

The next set of host region and the embedding watermark information is considered for completing the procedure. The algorithm comes to an end once all possible blocks are embedded.

Step 11. Reunite the watermarked image quarters Q'_j and output the watermarked image I_w.

The individually watermarked image quarters are rearranged in their original order to produce the watermarked image embedded with a watermark signal sized PxQ. $I_w = \{Q'_1, Q'_2, Q'_3, Q'_4\}$.

Algorithm 2: Watermark Embedding

```
Input: embedding region, $R_{sx} = \{R_{sR}, R_{sG}, R_{sB}\}$, Watermark bit w
for i = 1 to 3
    for j = 1 to 3
        if(w = = 1), $R'_{sB}(i, j) = R_{sB}(i, j) + M_1(i, j)$
        else
        $R'_{sB}(i, j) = R_{sB}(i, j) - M_1(i, j)$
    end j
end i
for x = {R, G}
    for i = 1 to 3
        for j = 1 to 3
            if(w = = 1), $R'_{sx}(i, j) = R_{sx}(i, j) + M_2(i, j)$
            else
            $R'_{sx}(i, j) = R_{sx}(i, j) - M_2(i, j)$
        end j
    end i
end x
Output: Watermarked regions, $R'_{sx}$
```

2.4.2. Watermark extraction algorithm

Step 1. Input the watermarked image I′ and image, I.

The watermarked image and its original is required as input to the extraction phase.

Step 2. Decompose I′ into four quarter regions, Q'_j and I into four quarter regions, Q_j, each of size M/2 × N/2.

Here *j* = 1, 2, 3, 4. The base image and the watermarked image are decomposed into four composite regions each of size M/2xN/2, thus generating four set of quarter regions.

Step 3. Consider non-overlapping 8x8 blocks, BQ_j and BQ'_j.

The region selected in I and I′ is decomposed into 8 × 8 blocks and the location used at the time of embedding is next identified to extract the watermark content. The total number of blocks BQ under one Q_j is M/16 * N/16.

Step 4. Apply algorithm 1 to locate the region of embedding.

The sub-region identification in the blocks from Q_j is carried out to determine the positions where the watermark bit was embedded during the embedding process. Block BQ_j is split into four sub-regions and the location of embedded R_{sB} is identified.

Step 5. Extract the watermark, W'_{iq} using Algorithm 3.

The sub-regions under blocks BQ_{jk} and BQ'_{jk} are analyzed using the procedure for extraction and the watermark bit hidden in the region is detected. Next, increase the block number k and watermark array index q and extract the next bit of watermark by repeating from step 4 until k == (M/16 * N/16).

Step 6. Continue extraction for subsequent quarter region Q_j and Q'_j by repeating from step 4 until *j* = 4.

After extracting one watermark sub-image, the above bit extraction steps are repeated for the remaining regions in the image.

Step 7. Combine the four extracted sub-images W'_i to output extracted watermark W′.

When the image extraction analysis is complete four set of watermarks will be obtained, W'_1, W'_2, W'_3 and W'_4. The size of each is M/16 × N/16. These are rearranged to output the final extracted watermark image of size P × Q.

Algorithm 3: Watermark Extraction

```
Input: Image region, $R_{sx} = \{R_{sR}, R_{sG}, R_{sB}\}$, $R'_{sx} = \{R_{sR}, R_{sG}, R_{sB}\}$
Initialize Ps = 0, P′s = 0
for x = {R, G,B}
    for i = 1 to 3
        for j = 1 to 3
            Ps = Ps + $R_{sx}(i, j)$
            P′s = P′s + $R'_{sx}(i, j)$
    end j
    end i
end x
 if(P′s > Ps)
 wb′ = 1 else
 wb′ = 0
Output: Watermark bit, wb′
```

3. Experimental analysis

The proposed watermarking technique is experimented on a large data base of color images. The host image used is a 24-bit color of size 512 × 512 and the watermark signal is a binary logo image. For testing a watermark of size 64 × 64 is used. In the initial watermark preprocessing stage the watermark is divided into four sub-images and each watermark sub-image is embedded in each of the four host image quadrant regions. This method is adopted to enhance the probability of watermark retrieval at the time of attacks that try to modify certain regions in the image or to impair the hidden watermark. Fig. 5 shows three host images utilized for carrying out the experimental analysis. The watermarked images generated were also forwarded to the extraction phase of the proposed algorithm to test the recoverability. Fig. 5g, (i) and (k) are the extracted watermark sub-image signals prior to recombining. Thus each image quarter successfully generates a miniature of the embedded watermark and finally the four were combined together to generate the 64 × 64 watermark.

3.1. Imperceptibility test

To evaluate the quality of the watermarked image generated by the proposed procedure two metrics are used, PSNR and SSIM. PSNR is the Peak Signal to Noise Ratio. This metric measures the amount of noise added to the actual contents during the process of watermarking. The watermark information added modifies the image and degrades the contents. An effective watermark embedding system should hence ensure that no significant distortions are introduced while watermark integration. The PSNR value drops as the level of distortion increases and hence a high PSNR value is regarded as one of the most desirable features of embedding process. Eq. (3) is used for computing PSNR and the Mean Squared Error (MSE) is computed using (4).

$$\mathrm{PSNR} = 10\log_{10}\frac{R * R}{\mathrm{MSE}} \tag{3}$$

$$\mathrm{MSE} = \sum_{k=1}^{3}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{(W(i,j) - I(i,j))}{3 * M * N} \tag{4}$$

here M × N is the size of the host image I, W is the watermarked image and R = 255 is the maximum intensity value used in image representation model.

The second metric used for evaluation is Structural Similarity Index Measure (SSIM) (Zhou et al., 2004). This measure examines

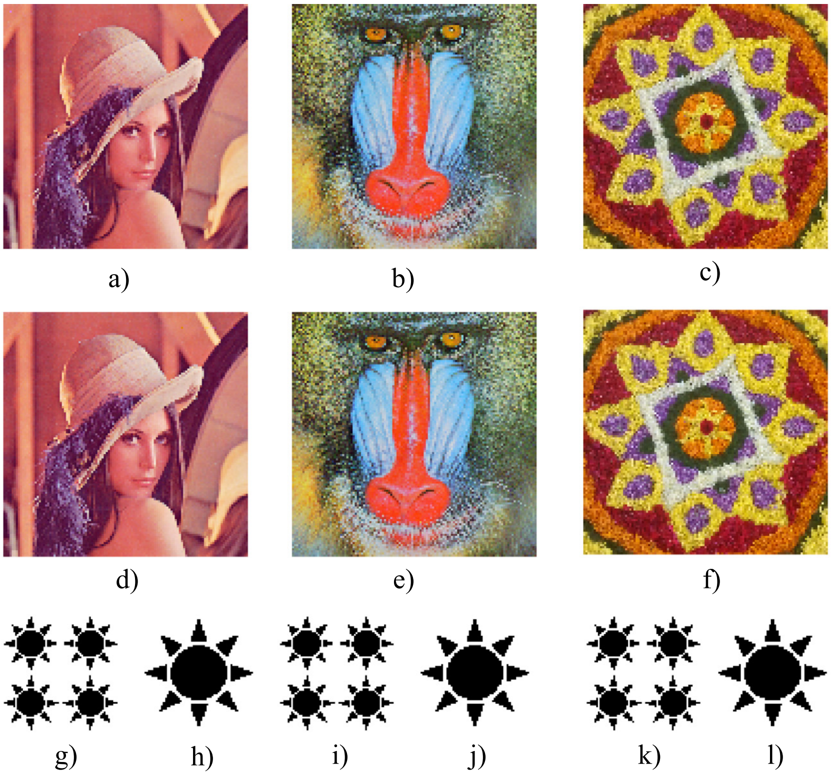


Figure 5. (a)–(c) Original test images, (d)–(f) watermarked images, (g), (i) and (k) extracted watermarks from each quarter, (h), (i) and (l) extracted watermark.

the distortion or change in a block-wise fashion. SSIM is computed using (5). SSIM measures the similarity between two images in a way closer to how human eyes perceive the image. It varies in the range $[-1\ 1]$, the maximum value of 1 indicates the two images are identical.

$$\mathrm{SSIM} = \mathrm{l(I,W).c(I,W).s(I,W)} \tag{5}$$

$$\mathrm{l(I,W)} = \frac{2\mu_I\mu_W + C_1}{\mu_I^2 + \mu_W^2 + C_1}$$

$$\mathrm{c(I,W)} = \frac{2\sigma_I\sigma_W + C_2}{\sigma_I^2 + \sigma_W^2 + C_2}$$

$$\mathrm{s(I,W)} = \frac{\sigma_{IW} + C_3}{\sigma_I\sigma_W + C_3}$$

here I is the host image and W is the watermarked image. l(I,W) is luminance comparison function, c(I,W) is contrast comparison and s (I,W) is structural comparison.

Another metric used is GEI (Global Embedding Impact). GEI is computed using (6). GEI varies from 0 to R (R = 255). This measure gives an estimate on the net variation a pixel in the image array has to undergo because of watermark embedding process in the base image. For example if an intensity value of v is added to all the pixels in the image then GEI measure is equal to v. The maximum variation a grayscale image pixel can undergo is 255, which is nothing but the maximum grayscale intensity value used for the image representation.

$$\mathrm{GEI} = \frac{\sum_{k=1}^{3}\sum_{i=1}^{M}\sum_{j=1}^{N}|(\mathrm{W(i,j)} - \mathrm{I(i,j)})|}{3 * \mathrm{M} \times \mathrm{N}} \tag{6}$$

Table 1 shows the PSNR and SSIM readings for different test images. Readings are observed for different payloads varying from 32 × 32 to 64 × 64 bits of watermark information. It can be seen that as payload increases the PSNR and SSIM values drop from their maximum values signaling an increase in the amount of noise addition is degrading the image quality. Table 2 lists GEI values that indicate each pixel is subjected to a variation 0.33 on the average. And as k is doubled the error distribution also is seen to increase by two folds.

3.2. Robustness to attacks

The ability of watermarked image to survive watermark removal attacks is further experimented. For testing this, watermarked images are subjected to various attacks and are forwarded

Table 1
PSNR and SSIM values.

Image	Payload									
	$1*(32\times32)$, k = 8		$2*(32\times32)$, k = 8		$3*(32\times32)$, k = 8		$4*(32\times32)$, k = 8		$4*(32\times32)$, k = 8/2	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Baboon	53.46	0.9969	50.63	0.9934	48.87	0.9918	47.62	0.9902	53.35	0.9974
Lena	53.64	0.9927	50.63	0.9934	48.89	0.9804	47.62	0.9725	53.35	0.9930
Flowers	53.65	0.9980	50.65	0.9956	48.89	0.9930	47.64	0.9904	53.36	0.9975

Table 2
GEI measurement.

	Lena	Baboon	Flower
K = 8	0.3333	0.3325	0.3333
K = 16	0.6666	0.6663	0.6666

to the watermark extraction phase for extracting the embedded signal. Depending upon the intensity of the attack the extracted signal may not be in the original form and hence it is compared with the original watermark signal to estimate the extent of deformation and possible loss in information. The better the extracted watermark image contents the higher the resilience to watermark removal attacks. Two commonly used metrics for robustness evaluation are Normalized Correlation Coefficient (NCC) and Bit error Ratio (BER).

NCC is computed using (7). Here, W is the original watermark signal and W′ is the extracted watermark image. When the two compared signals are identical the NCC value will be equal to 1. And as the variations between the two signals increases the NCC value drops and reads 0 when the two signals are fully dissimilar.

$$\mathrm{NCC} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[W(i,j)W'(i,j)]}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[W(i,j)]^2}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[W'(i,j)]^2}} \tag{7}$$

Bit Error Ratio is the measure of number of extracted bits in error versus the number of embedded bits. BER is computed using (8).

$$\mathrm{BER} = \frac{E_b}{T_b} \tag{8}$$

Here E_b is the count of error bits and T_b is the total number of embedded bits. Hence if all bits are extracted correctly, E_b equals 0 and BER will be zero. And as the error increases BER approaches 1 indicating the samples tested are dissimilar. Table 3 lists the observed NCC and BER values. The abbreviation B.C.E stands to indicate the count of Bits Correctly Extracted.

Figs. 6 and 7 show some of the attacked watermarked images along with the corresponding extracted watermark image. Fig. 7 shows cropped image and the extracted watermark image obtained from the remaining regions. During testing the image is cropped at different measures- 25%, 50% and 75%. And whenever a quarter region of the watermarked image is found the watermark extraction could proceed successfully to generate a sub-watermark. The miniature version of the watermark is recognizable enough to identify the owners of the digital resource.

Table 4 shows comparison of the proposed method against two methods presented in Huynh-The et al. (2016) and Liu (2012). We have selected two distinct watermarking schemes, the first method is frequency domain based and the second method is spatial domain based. The PSNR values obtained for the proposed scheme is much higher than the other two. The indicated value of 47.6 dB is obtained for watermarking color images using a payload of 64 × 64. Hence it can be concluded that the error variations introduced while watermark embedding is minimal and image degradation is lesser.

The proposed scheme generates high quality watermarked images that are reasonably robust to a variety of attacks. Table 3 illustrates the robustness of the algorithm to malicious attacks such as filtering and intentional LSB resetting. In majority of the existing spatial domain schemes if the modulated LSB bits are destroyed the watermark retrieval becomes unsuccessful. However in the proposed method we are able to recover the embedded watermark even after the LSB bits are distorted. Thus it can be concluded that the discussed algorithm survives several malicious watermark removal attacks and at the same time guarantees a fair PSNR value. The increased robustness can be attributed to the use of embedding mask M_1 for robustly integrating the watermark bit. And the use of embedding mask M_2 combined with the use of SIRD for local region selection for watermark embedding delivers imperceptibility and superior image quality.

Majority of the previous color image watermarking schemes use different methods for watermark addition. They either embed the watermark in all the three channels, R, G and B, equally or convert the image to another color space such as HSI or YC_bCr and employs the Y component to embed the watermark information. However as blue component is more insensitive to human eyes compared to R and G, more of B can be used for embedding the watermark signal. The proposed method considers R, G, B channels

Table 3
NCC and BER values.

Attack	**Lena**			**Baboon**			**Flowers**		
	B.C.E	**NCC**	**BER**	**B.C.E**	**NCC**	**BER**	**B.C.E**	**NCC**	**BER**
No Attack	4096	1	0	4096	1	0	4096	1	0
Salt & Pepper	3904	.9710	.0391	3930	.9680	.0449	3977	.9835	.0234
Poisson	3516	.9085	.1270	3515	.9137	.1201	3758	.9390	.0859
Speckle	3494	.9129	.1211	3463	.8986	.1406	3751	.9254	.1045
Average Filtering	3700	.9451	.0771	2946	.7991	.2686	3060	.8156	.2471
Gaussian LPF	4036	.9993	.0010	3851	.9563	.0615	3983	.9827	.0244
Sharpening	3744	.9455	.0781	3073	.7888	.3008	3197	.8450	.2246
JPG *(QF: 80)*	3332	.8648	.1875	2803	.7408	.3389	3092	.7985	.2754
JPG *(QF: 70)*	3168	.8314	.2295	2726	.7260	.3535	2995	.7729	.3029
JPG *(QF: 60)*	2992	.8057	.2617	2643	.7241	.3525	2792	.7448	.3389
JPG *(QF: 50,k = 16)*	3453	.8873	.1543	3047	.7909	.2744	3240	.8332	.2256
JPG *(QF: 40,k = 16)*	3280	.8549	.1982	2986	.7842	.2852	3130	.8128	.2529
JPG *(QF: 30,k = 16)*	3100	.8199	.2422	2849	.7736	.2979	3008	.7923	.2705
JPG *(QF: 20,k = 16)*	2738	.7362	.3506	2684	.7427	.3350	2786	.7396	.3398
Cropping 25%	3702	.7500	.2500	3702	.7500	.2500	3702	.7500	.2500
Cropping 50%	2048	.5000	.5000	2048	.5000	.5000	2048	.5000	.5000
Cropping 75%	1024	.2500	.7500	1024	.2500	.7500	1024	.2500	.7500
LSB reset B(1or2)	4096	1	0	4096	1	0	4096	1	0
LSB reset B(1–3)	4096	1	.0117	4067	.9959	.0059	4025	.9857	.0205
LSB reset B(1–4)	3593	.9205	.1104	3653	.9214	.1094	3583	.9152	.1191
Resizing 50%	3845	.9633	.0518	3081	.8188	.2441	3250	.8374	.2187
Resizing 50% *(k = 16)*	4021	.9917	.0117	3545	.8951	.1445	3748	.9375	.0869

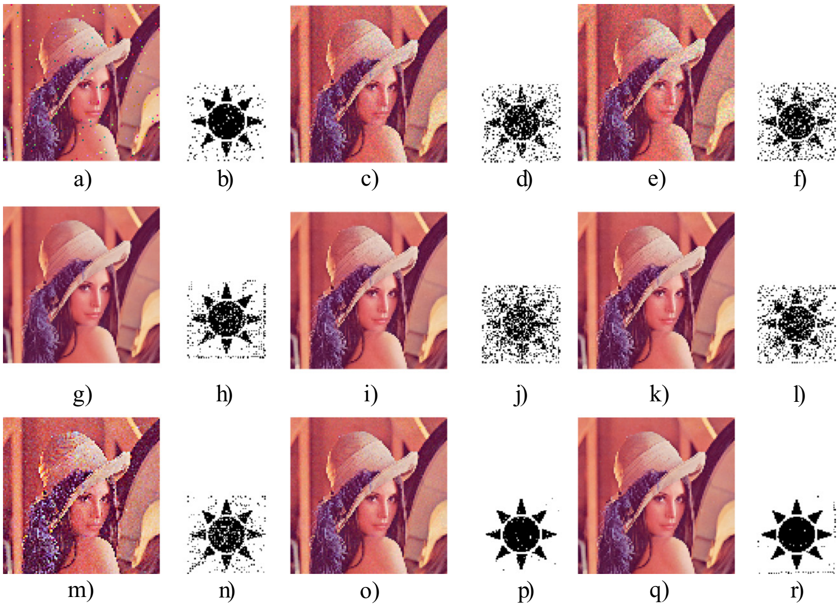


Figure 6. Watermarked image attacked by: – Salt & Pepper(a), Poisson(c), Speckle(e), Average filtering(g), JPG(70)(i), JPG(50,sf = 2)(k), Sharpening(m), LSB reset(o), and Gaussian LPF (p). Corresponding extracted watermark, (b), (d), (f), (h), (j), (l), (n), (p) and (r).

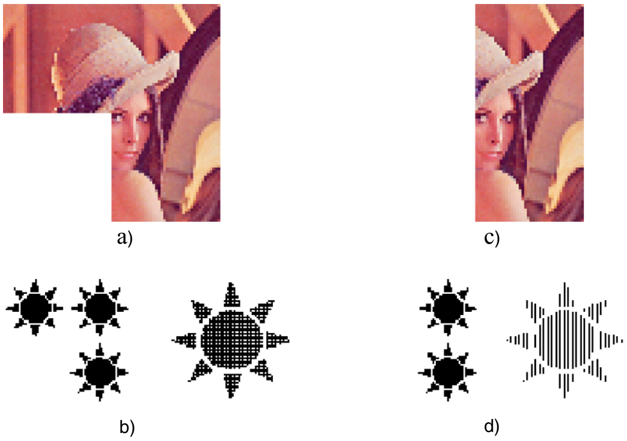


Figure 7. (a) and (b) Cropped watermarked image (25%) and extracted watermark, (c) and (d) cropped watermarked image (50%) and extracted watermark.

Table 4
Comparison between the proposed method and methods by Huynh-The et al. (2016) and Liu (2012).

	Thien Huynh-The et al.	Liu K. C	Proposed method
Host image size	512 × 512	512 × 512	512 × 512
Image type	Color	Color	Color
Watermark	64 × 64	Image Features	64 × 64
Operating domain	Frequency Domain	Spatial Domain	Spatial Domain
Technique used	Four-level DWT	LSB Replacement	LSB Modification
Speed of operation	Low	Low	Fast
Embedding color domain	RGB	RGB	RGB
Embedding Color component	R, G, B	R, G, B	B
PSNR (dB)	43.51	39	47.6
Signal quality	Good	Good	Superior
Adjustments for color variation	No	No	Yes

differently and different measures are used for integrating the identification watermark in individual color channels so as to keep the distortion level minimal.

4. Conclusion

The paper proposes an algorithm for embedding a logo image in a color image. The two recommended masks, embedding mask M_1 and compensation mask M_2 ensure the embedded bits are less distracting to the human eyes. Also as the watermark information is spread to wider areas the chances for watermark survival to attacks are enhanced. When the pixels are modified independent of their neighbors at pixel-level, as in conventional spatial domain schemes, there is a high probability of creation of salt & pepper noise effect. In the proposed method the usage of masks ensure that the modified pixels do not stand out compared to the pixels in the neighborhood. Mask M_2 is used as a compensation mask to ensure that the original color distributions are least affected while embedding. The proposed algorithm is tested on numerous images. The robustness and imperceptibility of the watermarked images were experimentally evaluated. The proposed scheme delivers high quality of watermarked images that can survive a good number of attacks.

References

Bedi, S.S., Rakesh, Ahuja, Himanshu, Agarwal, 2013. Copyright protection for video watermarking based on wavelet transform in multiband. Int. J. Comput. Appl. 66 (8).

Chih-Chin, Lai, Cheng-Chih, Tsai, 2010. Digital image watermarking using discrete wavelet transform and singular value decomposition. IEEE Trans. Instrum. Meas. 99 (11).

Chun-Hung, Chen, Yuan-Liang, Tang, Chih-Peng, Wang, Wen-Shyong, Hsieh, 2014. A robust watermarking algorithm based on salient image features. International Journal of light and electron optics (Optik) 125, 1134–1140.

Himeur, Yassine, Boudraa, Bachir, Khelalef, Aziz, 2012. A secure and high robust audio watermarking system for copyright protection. Int. J. Comput. Appl. 53 (17).

Huynh-The, Thien, Banos, Oresti, Lee, Sungyoung, Yoon, Yongik, Le-Tien, Thuong, 2016. Improving digital image watermarking by means of optimal channel selection. Expert Syst. Appl. 62, 177–189.

Junpeng, Zhang, Qingfan, Zhang, Hongli, Lv, 2013. A novel image tamper localization and recovery algorithm based on watermarking technology. Int. J. Light Electr. Optics (Optik) 124, 6367–6371.

Keqiang Ren, Huihuam Li, 2011, Large Capacity Digital Audio Watermarking Algorithm based on DWT and DCT. In: International Conference on Mechatronic Science, Electrical Engineering and Computer, Jilin, China, 2011.

Lin, Phen Lan., Chung-Kai, Hsieh, Po-Whei, Huang, 2005. A hierarchical digital watermarking method for image tamper detection and recovery'. Pattern Recogn. 38, 2519–2529.

Liu, K.C., 2012. Color image watermarking for tamperproofing and pattern based recovery. IET Image Proc. 6 (5), 445–454.

Manikanda Prabu, S., Ayyasamy, Dr.S., 2014. An efficient watermarking algorithm based on DWT and FFT approach. Int. J. Comput. Sci. Eng.

Mei, Yu, Wang, Jing, Jiang, Gang, Peng, Zongju, Shao, Feng, Luo, Ting, 2015. New fragile watermarking method for stero image authentication with localization and recovery. Int. J. Electron. Commun. 69, 361–370.

Parthasarathy, Arvind Kumar, Subhash, Kak, 2007. An improved method of content based image watermarking. IEEE Trans. Broadcast. 53 (2).

Quing Liu, Jun Ying, 2012, Grayscale Image Digital Watermarking Technology based on Wavelet Analysis, IEEE Symposium on Electrical & Electronics Engineering pp 618–621, 2012.

Quingtang, Su, Niu, Yugang, Wang, Qingjun, Sheng, Guorui, 2013a. A novel blind digital watermarking for embedding color image into color image. Int. J. Light Electr. Optics (Optik) 124, 6255–6260.

Quingtang, Su, Niu, Yugang, Liu, Xianxi, Yao, Tao, 2013b. A blind color image watermarking based on DC component in the spatial domain. Int. J. Light Electr Optics (Optik) 124, 3254–3259.

Razieh, Keshavarzian, Ali, Aghagolzadeh, 2016. ROI based robust and secure image watermarking using DWT and Arnold map. Int. J. Electron. Commun. 70, 278–288.

Sajjad, Dadkhah, azizah abd, Manaf, Yoshiaki, Hori, Aboul, Ella Hassanien, Somayeh, Sadeghi, 2014. An effective SVD-based image tampering detection and self-recovery using active Watermarking'. Signal Process.: Image Commun. 29, 1197–1210.

Shiguo, Lian, Dimitris, Kanellopoulos, Giancarlo, Ruffo, 2009. Recent advances in multimedia information system security. Informatica 33.

Vidhyasagar M. Potdar, Song Van, E. Chang, 2005, A Survey of Digital Image watermarking Techniques, Third IEEE International Conference on Industrial Informatics, pp 709–716, 2005.

Zhicheng, Ni, Yun-Qing, Shi, Nirwan, Ansari, Su, Wei, 2006. Reversible data hiding. IEEE Trans. Circuits Syst. Video Technol. 16 (3), 354–362.

Zhou, Wang, Alan Conrad, Bovik, Hamid Rahim, Sheikh, Simoncelli, Eero P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13 (4).