Int. J. Electron. Commun. (AEÜ) 67 (2013) 102–112

Contents lists available at SciVerse ScienceDirect

International Journal of Electronics and Communications (AEÜ)

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

Robust blind image watermarking scheme based on Redundant Discrete Wavelet Transform and Singular Value Decomposition

Nasrin M. Makbol*, Bee Ee Khoo

School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Malaysia

ARTICLE INFO

Article history:
Received 27 December 2011
Accepted 28 June 2012

Keywords:
Image watermarking
Singular Value Decomposition
Redundant Discrete Wavelet
Geometrical attacks
Imperceptibility

ABSTRACT

Copyright protection and proof of ownership are two of the main important applications of the digital image watermarking. The challenges faced by researchers interested in digital image watermarking applications lie in the creation of new algorithms to serve those applications and to be resistant to most types of attacks, especially the geometrical attacks. Robustness, high imperceptibility, security, and large capacity are four essential requirements in any watermarking scheme. This paper presents a new image watermarking scheme based on the Redundant Discrete Wavelet Transform (RDWT) and the Singular Value Decomposition (SVD). The gray scale image watermark was embedded directly in the singular values of the RDWT sub-bands of the host image. The scheme achieved a large capacity due to the redundancy in the RDWT domain and at the same time preserved high imperceptibility due to SVD properties. Embedding the watermarking pixel's values without any modification inside the wavelet coefficient of the host image overcomes the security issue. Furthermore, the experimental results of the proposed scheme showed a high level of robustness not only against the image processing attacks but also against the geometrical attacks which are considered as difficult attacks to resist.

1. Introduction

Recently, the vast increase in the pirated digital media due to the world wide web availability and the speed of distribution has led to the need to protect the media against attacks. The digital media (such as video, image, audio or text) can be modified easily by attackers who can then claim its ownership. So, owners, authors, publishers and providers of that media, are reluctant to grant the distribution of their documents in a networked environment [1]. The need to develop robust methods to protect the intellectual property rights of data owners against unauthorized copying, and redistributing it on the network became the main objective of researchers in digital watermarking. Traditional methods such as copy protection or encryption could not solve the problem of unauthorized copying entirely. Digital watermarking presents a viable solution to that problem by marking the digital media, then it can be easy to be spread and later track it [1]. This can be referred to as a digital signature. The technology used to apply the digital signature on the digital media is called copyright protection.

Digital watermarking is described as technologies and methods that hide information sometimes called a signal or watermark; for example, a number or text, into media files such as images, videos, audio, and text. A digital watermark can be visible or invisible to the human visual system. Logos that are often seen added to the corners of images or videos as a way to prevent copyright infringement is an example of visible digital watermark. These watermarks can be easily defeated and removed by replacing or cropping it from the digital media.

For better security, the watermark has to be invisible to the human eye to ensure that the attackers cannot access the secured data and remove or alter them. The system has to take into account the data quality after embedding the watermark. Good data quality means high system imperceptibility. The robustness is the ability of the system to maintain the validity of the inclusion of a watermark even after any attack such as geometrical or non-geometrical attacks. Watermark capacity is considered as a challenge in watermarking systems. It refers to the number of bits that can be embedded in an image, video, etc. [2]. Increasing the capacity of the embedded data will improve the system robustness but it may affect the media imperceptibility. A trade off always exists between the imperceptibility and the robustness. Nevertheless, they are considered as the most important requirements for designing any watermarking scheme, especially for copyright protection purposes. It is important to compromise between these issues [1].

As a result, the recent effort of research intends to develop watermarking systems to protect the media content. These systems should satisfy the imperceptibility, the robustness, the security and the capacity requirements.

* Corresponding author. Tel.: +604-5996032; fax: +604-5941023.
E-mail addresses: Nasrin.ld08@student.usm.my (N.M. Makbol), Beekhoo@eng.usm.my (B.E. Khoo).

1434-8411/$ – see front matter © 2012 Elsevier GmbH. All rights reserved.
http://dx.doi.org/10.1016/j.aeue.2012.06.008

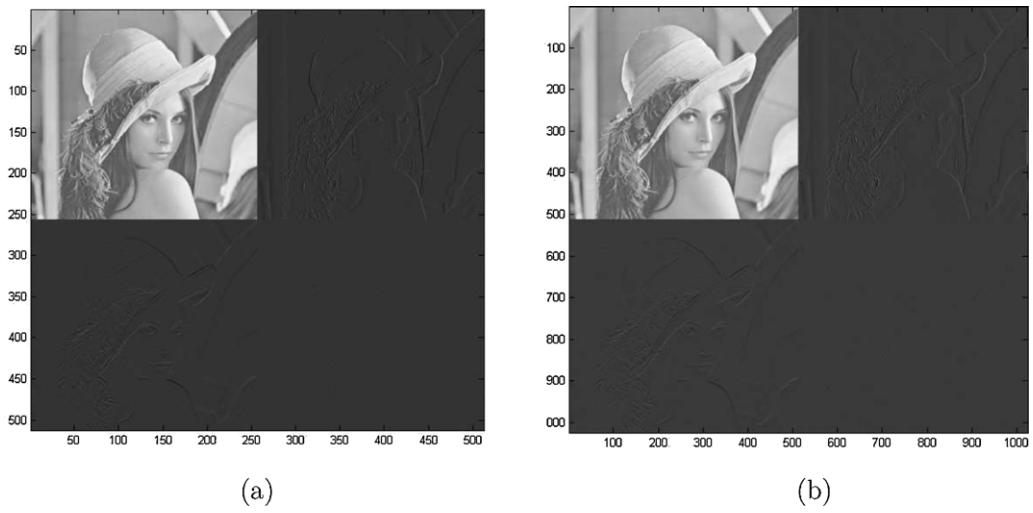

Fig. 1. (a) One level Haar DWT (the image is 512 × 512 pixels and each sub-band is 256 × 256 pixels); (b) one level Haar RDWT (the image is 512 × 512 pixels and each sub-band is 512 × 512 pixels).

In the World Wide Web, applying watermarks onto the digital media will serve various purposes such as authentication, copyright control, and copyright protection. According to the applications and purposes of the watermarking technique [3], the digital watermark techniques are categorized as robust, fragile, or semi-fragile. Whereas according to the domain in which the watermark will be embedded, it is categorized into spatial domain techniques and transform domain techniques. In the spatial domain techniques, the watermark will be embedded by directly changing the pixel values of the media data. In the transform domain techniques, the embedding process will take place by altering the transform coefficients. Although the embedding in the spatial domain is not extensive in time and computation but it will not resist against the geometric attacks. In contrast, the embedding in the transform domain will ensure a good robustness against image processing attacks and a lot of geometric attacks. Finite Ridgelet Transform (FRIT) [4], Discrete Cosine Transform (DCT) [5], Discrete Fourier Transform (DFT) [6], Discrete Wavelet Transform (DWT) [7], Redundant Discrete Wavelet Transform (RDWT) [8] and the Singular Value Decomposition (SVD) [7,9] are the most common examples of transform domain techniques.

Recently, some researchers have suggested the use of two or three transforms to develop their watermarking schemes. Their aim was to improve the robustness and achieve high imperceptibility. These schemes are called hybrid watermarking schemes [10,9,7,11,12].

The SVD transform always shows a good performance in the new developed watermarking schemes. Its primary advantage lies in keeping minor changes with largest changed singular values which can be occurred by the attacks. Furthermore, the SVD is preferred to be implemented together with other transforms. This is because it requires extensive computations if applied separately onto images. Ganic et al. [11] inserted the singular values of the gray scale watermark into the singular values of all one-level DWT sub-bands. Rastegar et al. [9] also suggested a hybrid watermarking scheme. They applied radon transform to the host image, then decomposed it into three levels using DWT. The singular values of the third-level DWT sub-bands are modified by the singular values of the binary watermark. Lagzian et al. [8] replaced DWT by RDWT and followed the same steps used by Ganic et al. [11]. These schemes performed well against some attacks, but their schemes have a weakness in the security because they applied SVD for the watermark in the embedding process [13,14,12]. Lai et al. [7] scheme could overcome this security issue. They decomposed the image using DWT, divided the watermark into two halves and then embedded each half into the singular values of LH and HL sub-bands of DWT transform of the host image. Although, Lai et al. scheme fulfilled the requirements of digital image watermarking, their scheme's capacity is still insufficient. To satisfy all watermarking requirements, we present a new hybrid image watermarking scheme in this paper. The proposed scheme is based on Lai et al. [7], Ganic et al. [11], and Lagzian et al. [8]. The gray scale watermark image is embedded in the singular values of all RDWT sub-bands. First, the RDWT is used to decompose the host image into four sub-bands. Each sub-band size was similar to the host image size. Then, the sub-bands singular values are modified directly by the watermark pixel values. This is to achieve the capacity requirement. The proposed scheme achieved a high level of robustness not only against the image processing attacks but also against the geometrical attacks. The scheme results showed high imperceptibility and security.

This paper is organized as follows. The RDWT and SVD transform are briefly described in Sections 2 and 3, respectively. Then, in Section 4 the proposed scheme is illustrated. In Section 5, the experimental results are presented. In Section 6, a comparative analysis of our scheme and other schemes is given, and finally the conclusions are presented.

2. Redundant Discrete Wavelet Transform (RDWT)

DWT is very useful to determine areas in the original image where a watermark can be imperceptibly inserted because of its excellent spatio-frequency localization properties. So, it is commonly used for watermarking. On the other hand, DWT has a lot of disadvantages. The major disadvantage is the shift variant. This occurs due to the downsampling process after each level of filtering, which causes a significant change in the wavelet coefficients of the image even for minor shifts in it. This leads to inaccurate extraction of the watermark data and the cover image. RDWT is established to overcome that problem because it is shift invariant.

RDWT has been independently discovered a number of times and was given a number of different names, including the algorithm *áa* trous, the overcomplete DWT (ODWT), the undecimated DWT (UDWT), the discrete wavelet frames (DWF) and the shift-invariant DWT (SIDWT) [15]. There are multiple methods to implement the RDWT, and multiple methods to represent the resulting overcomplete set of coefficients. The original implementation was in a form of the algorithm *áa* trous, which, in its core, eliminates the downsampling operator from the usual implementation of the DWT [15].

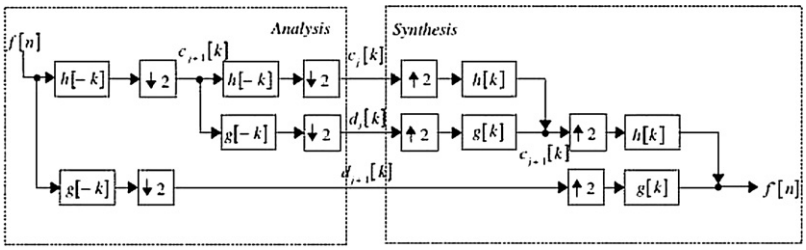

Fig. 2. Two level 1D DWT analysis and synthesis filter banks.

For any image, important textures in it will be at the same spatial location in each sub-band, if each sub-band maintains the same size as the original image. In DWT, the sizes of each sub-band is decreased when the decomposition level is increased. This can be seen in Fig. 1. Thus, for DWT as decomposition level increases, it should consider increasingly larger spatial regions. But this can not be achieved in DWT technique. Based on that observations, more accurate capturing of the local texture within RDWT domain can be done [16]. So, the exact measure of local texture can be used. It is used to steer the watermark casting, and increase the watermark capacity into those areas with advanced texture where the human visual system (HVS) is less sensitive to embedded watermark. A 1D DWT and its inverse are shown in Fig. 2 [16] ($f[n]$ represents the 1D input signal and $f'[n]$ represents the reconstructed signal). The lowpass and highpass analysis filters are $h[-k]$ and $g[-k]$ and the corresponding lowpass and highpass synthesis filters are $h[k]$ and $g[k]$. c_j and d_j are the low-band and high-band output coefficients at the level j.

1. DWT analysis.

$$c_j[k] = (c_{j+1}[k] * h[-k]) \downarrow 2, \tag{1}$$

and

$$d_j[k] = (c_{j+1}[k] * g[-k]) \downarrow 2. \tag{2}$$

where * indicates convolution, and ↓2 indicates downsampling. That is, if $y[n] = x[n] \downarrow 2$, then

$$y[n] = x[2n]. \tag{3}$$

2. DWT synthesis

$$c_{j+1}[k] = ((c_j[k] \uparrow 2) * h[k] + (d_j[k] \uparrow 2) * g[k]), \tag{4}$$

↑2 here denotes to the upsampling process. That is, if $y[n] = x[n] \uparrow 2$, then

$$y[n] = \begin{cases} x\left[\frac{n}{2}\right], & n \text{ even}, \\ 0, & n \text{ odd}. \end{cases} \tag{5}$$

Fig. 3 shows a 1D RDWT and its inverse transform [16]. The RDWT removes downsampling and upsampling of coefficients. At each level, the number of output coefficients doubles that of the input. Therefore, the filters themselves are upsampled to suit the length of the growing data. Precisely, the filters for scale j are [16]:

$$h_j[k] = h_{j+1}[k] \uparrow 2, \tag{6}$$

and

$$g_j[k] = g_{j+1}[k] \uparrow 2. \tag{7}$$

1. RDWT analysis.

$$c_j[k] = (c_{j+1}[k] * h_j[-k]), \tag{8}$$

and

$$d_j[k] = (c_{j+1}[k] * g_j[-k]). \tag{9}$$

2. RDWT synthesis

$$c_{j+1}[k] = \frac{1}{2}(c_j[k] * h_j[k] + d_j[k] * g_j[k]). \tag{10}$$

Eqs. (6)–(10) are known as the algorithm *áa trous*, since the filter-upsampling procedure embeds "holes" ("trous" in French) between the filter taps.

3. Singular Value Decomposition (SVD)

SVD is a technique developed for a wide variety of applications. It is used in signal and image processing applications such as image hiding, image compression, noise reduction and image watermarking. Several watermarking approaches have been proposed based on the SVD. Due to the extensive computations when SVD is applying alone on an image [7], some hybrid SVD-based algorithms have been developed such as DWT-SVD [7,11], and RDWT–SVD [8]. In these algorithms, the watermark is embedded in the transform coefficients of the host image instead of the pixel values. SVD is a numerical analysis tool used to analyze matrices. A matrix in SVD will be decomposed into three matrices. If A is an image, it is indicated as $A \in R^{nm}$, R represents the real number domain. Then, the SVD of A is defined as follows [17]:

$$A = USV^T \tag{11}$$

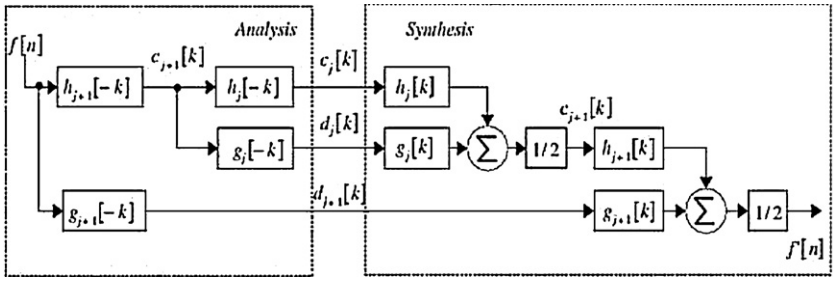

Fig. 3. Two level 1D RDWT analysis and synthesis filter banks.

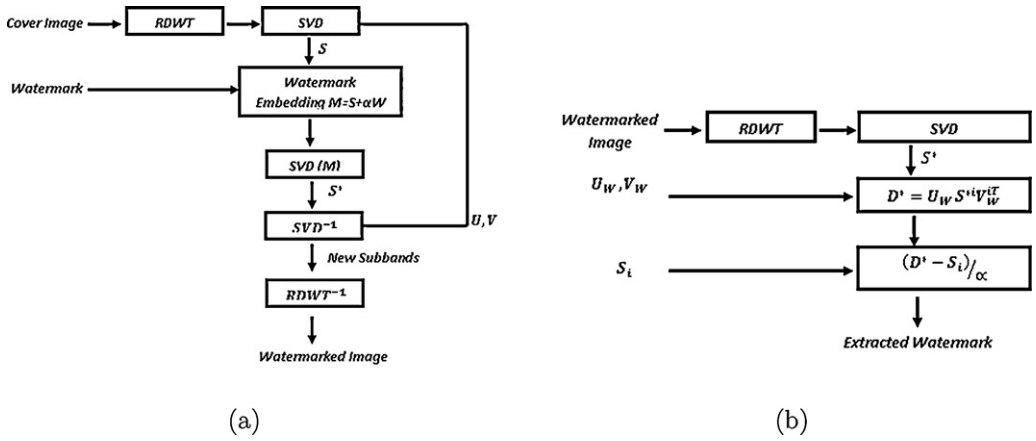

Fig. 4. (a) The process of watermark embedding; (b) the process of watermark extracting.

Fig. 5. (a) Host image (512 × 512 Peppers); (b) host image (512 × 512 Lena); (c) host image (512 × 512 Baboon); (d) gray watermark (512 × 512 Cameraman).

U and V are orthogonal matrices where $U, V \in R^{nm}$, S represents the diagonal matrix $S \in R^{nm}$ as shown below:

$$S = \begin{bmatrix} \sigma_{1,1} & & & \\ & \sigma_{2,2} & & \\ & & \ddots & \\ & & & \sigma_{n,n} \end{bmatrix}$$

Singular values are the diagonal elements $(\sigma_1, \ldots, \sigma_n)$ of the matrix S. They satisfy: $\sigma_{1,1} \geq \sigma_{2,2} \cdots \geq \sigma_{r,r} \cdots \geq \sigma_{n,n}$

Advantages of using SVD in the digital image processing field are as follows [17]:

1. Singular values, SVs, represent algebraic image properties, where S represents the brightness of the image while U and V are the geometry properties of the image.
2. Singular values of each image have a good stability. It means that for a small perturbation added to an image, its SVs do not change fast.
3. SVD matrices can be square or rectangle.

4. The proposed RDWT–SVD watermarking scheme

In this work, Redundancy Discrete Wavelet Transform (RDWT) is used together with Singular Value Decomposition (SVD). The procedure of RDWT–SVD watermark embedding and extraction algorithms are depicted in Fig. 4(a) and (b), respectively.

Fig. 6. (a) Watermarked Pepper image (*PSNR* 54.1556 db); (b) watermarked Lena image (*PSNR* 54.0353 db); (c) watermarked Baboon image (*PSNR* 55.9768 db).

Fig. 7. (a) Watermarked image under Speckle noise (0.4) (*PSNR* 10.6832 db); (b) watermarked image under Gaussian noise (0, 0.005) (*PSNR* 23.006 db); (c) watermarked image under Pepper & salt noise (0.001) (*PSNR* 35.5595 db); (d) watermarked image under median filter (3 × 3) (*PSNR* 35.4997 db); (e) watermarked image under JPEG compression Q = 40 (*PSNR* 35.0824 db); (f) watermarked image under HE (*PSNR* 19.1347 db); (g) watermarked image under rotation (angle 50°) (*PSNR* 10.36672 db); (h) watermarked image under shift 2% (*PSNR* 14.2681 db); (i) watermarked image under cut attack (*PSNR* 15.1333 db).

4.1. Watermark embedding algorithm

The steps are as follows:

- Perform 1-level RDWT onto the cover image to decompose it into its four sub-bands, which are LL, LH, HL and HH.
- Apply SVD to all sub-bands (LL, HL, LH, HH), as follows:

$$A^i = U^i S^i V^{iT} \tag{12}$$

where i indicates to the sub-bands (i.e., LL, LH, HL, HH).

- Modify the Singular Values S^i for each sub-band by embedding the watermark directly, and then apply SVD to them, respectively, as follows:

$$S^i + \alpha W = U_W^i S_W^i V_W^{iT} \tag{13}$$

where i indicates to the sub-bands (i.e., LL, LH, HL, HH) and α represents the scaling factor ($\alpha = 0.05$, for watermark embedding into LL sub-band, and $\alpha = 0.005$, for embedding into others sub-bands (HL, LH, HH)).

- Perform the new modified RDWT coefficients for each sub-band.

$$A^{inew} = U^i S_W^i V^{iT} \tag{14}$$

where i indicates to the sub-bands (i.e., LL, LH, HL, HH).

- Finally, apply the inverse RDWT using the four sets of the modified RDWT coefficients to obtain the watermarked image A_W.

$$A_W = RDWT^{-1} \tag{15}$$

4.2. Watermark extraction algorithm

The steps are as follows:

- Apply RDWT on the watermarked image A_W^* (possibly distorted) to decompose it into four sub-bands LL, LH, HL, HH.
- Apply SVD to all sub-bands (LL, HL, LH, HH), as follows:

$$A_W^* = U^{*i} S^{*i} V^{*iT} \tag{16}$$

where i indicates to the sub-bands (i.e., LL, LH, HL, HH).

- Compute

$$D^{*i} = U_W^i S^{*i} V_W^{iT} \tag{17}$$

where i indicates to the sub-bands (i.e., LL, LH, HL, HH).

- Compute

$$W^{*i} = \frac{D^{*i} - S^i}{\alpha} \tag{18}$$

where W^{*i} is the extracted watermark in level i.

Table 1
The *NC* of the extracted watermark under different attacks.

Attack	Pepper image				Lena image				Baboon image			
	LL	LH	HL	HH	LL	LH	HL	HH	LL	LH	HL	HH
Gamma correction attack 0.8	0.995	0.9950	0.9949	0.9836	0.995	0.9944	0.9950	0.9284	0.994	0.9929	0.9928	0.9932
Pepper & salt noise (density 0.3)	0.675	0.8607	0.8280	0.7979	0.668	0.8917	0.8926	0.8110	0.765	0.8983	0.8671	0.8594
Pepper & salt noise (density 0.01)	0.957	0.8669	0.8207	0.8668	0.950	0.8618	0.8212	0.8315	0.989	0.9593	0.9644	0.9902
Pepper & salt noise (density 0.001)	0.994	0.9768	0.9668	0.9342	0.994	0.9618	0.9689	0.8727	0.998	0.9893	0.9895	0.9892
Pepper & salt noise (density 0.005)	0.978	0.8954	0.8574	0.8949	0.974	0.8634	0.8548	0.8446	0.992	0.9792	0.9788	0.9903
Speckle noise (var = 0.01)	0.965	0.8563	0.8251	0.8695	0.952	0.8477	0.8103	0.8166	0.989	0.9551	0.9592	0.9890
Speckle noise (var = 0.04)	0.895	0.8826	0.8433	0.8415	0.873	0.9104	0.8688	0.8326	0.964	0.8829	0.9307	0.9595
Speckle noise (var = 0.4)	0.714	0.8574	0.8324	0.7944	0.704	0.8908	0.8980	0.8122	0.808	0.8988	0.8668	0.8457
Gaussian noise ($M = 0$, var = 0.001)	0.984	0.9171	0.8892	0.8659	0.979	0.8666	0.8823	0.7836	0.992	0.9840	0.9831	0.9897
Gaussian noise ($M = 0$, var = 0.005)	0.937	0.8454	0.8132	0.8472	0.925	0.8617	0.8156	0.8230	0.983	0.9337	0.9497	0.9872
Gaussian noise ($M = 0$, var = 0.5)	0.642	0.8488	0.8277	0.7939	0.636	0.8869	0.8935	0.8087	0.732	0.8832	0.8265	0.8073
Rotation (angle 45°)	0.982	0.9621	0.9460	0.8122	0.983	0.9823	0.9815	0.8045	0.931	0.9042	0.9679	0.9662
Rotation (angle 2°)	0.975	0.8608	0.8594	0.8827	0.981	0.8981	0.9327	0.9389	0.952	0.9592	0.9231	0.9863
Rotation (angle 70°)	0.979	0.9687	0.9300	0.8299	0.981	0.9668	0.9436	0.8619	0.921	0.9251	0.9331	0.9805
Rotation (angle 110°)	0.979	0.9416	0.9459	0.8414	0.981	0.9686	0.9202	0.9007	0.921	0.9210	0.9277	0.9825
Rotation (angle −50°)	0.983	0.9734	0.9345	0.8357	0.984	0.9751	0.9376	0.7810	0.931	0.8928	0.9689	0.9720
Rotation (angle 50°)	0.979	0.9707	0.9522	0.8407	0.985	0.9850	0.9724	0.7529	0.933	0.8981	0.9757	0.9713
Histogram Equalization	0.991	0.9893	0.9862	0.9910	0.990	0.9865	0.9848	0.9084	0.976	0.9723	0.9737	0.9757
JPEG compression $Q = 5$	0.951	0.6398	0.6945	0.6551	0.952	0.7666	0.7190	0.7487	0.915	0.5872	0.5676	0.5023
JPEG compression $Q = 10$	0.968	0.7513	0.7802	0.7725	0.972	0.8557	0.8328	0.7901	0.961	0.7496	0.7275	0.5908
JPEG compression $Q = 20$	0.979	0.8252	0.8425	0.8616	0.983	0.9120	0.8929	0.8275	0.984	0.8855	0.8696	0.7148
JPEG compression $Q = 30$	0.982	0.8562	0.8753	0.8822	0.987	0.9327	0.9223	0.8494	0.989	0.9385	0.9285	0.8085
JPEG compression $Q = 40$	0.985	0.8801	0.8962	0.8995	0.988	0.9411	0.9355	0.8653	0.991	0.9548	0.9480	0.8632
JPEG compression $Q = 50$	0.987	0.8999	0.9075	0.9028	0.990	0.9487	0.9470	0.8608	0.993	0.9610	0.9579	0.9033
JPEG compression $Q = 60$	0.989	0.9200	0.9231	0.9217	0.992	0.9564	0.9578	0.8734	0.994	0.9667	0.9707	0.9524
JPEG compression $Q = 70$	0.991	0.9353	0.9409	0.9186	0.994	0.9644	0.9670	0.8863	0.996	0.9837	0.9887	0.9868
JPEG compression $Q = 80$	0.994	0.9667	0.9729	0.9036	0.995	0.9746	0.9786	0.8960	0.998	0.9898	0.9890	0.9897
JPEG compression $Q = 90$	0.999	0.9927	0.9901	0.8915	0.997	0.9887	0.9896	0.9208	0.999	0.9895	0.9898	0.9904
JPEG compression $Q = 100$	0.999	0.9931	0.9930	0.9814	0.998	0.9922	0.9934	0.9176	0.998	0.9954	0.9949	0.9950
Median filtering (3×3)	0.983	0.9156	0.9344	0.9717	0.982	0.9581	0.9663	0.8840	0.924	0.9618	0.9199	0.958
Median filtering (5×5)	0.967	0.8710	0.8937	0.8955	0.950	0.9130	0.9268	0.9845	0.775	0.8953	0.8190	0.9472
Wiener filtering (3×3)	0.978	0.8782	0.8808	0.9333	0.984	0.9509	0.9609	0.9332	0.945	0.9191	0.8801	0.8499
Wiener filtering (5×5)	0.971	0.8681	0.8789	0.9362	0.974	0.9404	0.9561	0.9427	0.877	0.8997	0.8544	0.7750
Gaussian filtering (3×3)	0.987	0.9385	0.9376	0.8899	0.987	0.9652	0.9608	0.9417	0.977	0.9658	0.9515	0.9782
Gaussian filtering (5×5)	0.987	0.9383	0.9373	0.8901	0.987	0.9649	0.9606	0.9418	0.977	0.9657	0.9512	0.9781
Scaling (zoomout = 0.5, zoomin = 2)	0.949	0.6719	0.7038	0.8941	0.948	0.8094	0.7467	0.8605	0.871	0.6061	0.5880	0.5661
Scaling (zoomout = 0.25, zoomin = 4)	0.818	0.4994	0.5164	0.7545	0.788	0.5588	0.5137	0.8528	0.639	0.4880	0.4534	0.5004
Scaling (zoomout = 2, zoomin = 0.5)	0.992	0.9517	0.9544	0.8760	0.992	0.9772	0.9716	0.9412	0.987	0.9698	0.9539	0.9556
Sharpening	0.932	0.9047	0.8833	0.8932	0.927	0.9312	0.9237	0.9094	0.863	0.9138	0.9164	0.9275
Shift 2%	0.993	0.9558	0.9526	0.9973	0.9912	0.9804	0.989	0.8833	0.9925	0.9808	0.9722	0.9860
Shearing ($x = 1$, $y = 0.2$)	0.979	0.9277	0.9729	0.8875	0.980	0.9554	0.9862	0.8752	0.938	0.9860	0.9708	0.9681
Shearing ($x = 0.2$, $y = 1$)	0.973	0.9087	0.8773	0.9206	0.957	0.8621	0.7274	0.7813	0.851	0.9605	0.8519	0.7004
Cut 10	0.994	0.9905	0.9933	0.9827	0.994	0.9833	0.9928	0.9228	0.992	0.9906	0.9890	0.9913
Cut 20	0.989	0.9841	0.9911	0.9830	0.989	0.9820	0.9900	0.9137	0.990	0.9872	0.9856	0.9887
Cut 30	0.979	0.9833	0.9880	0.9804	0.986	0.9780	0.9865	0.9112	0.986	0.9825	0.9841	0.9850
Translate (10, 10)	0.993	0.9923	0.9929	0.9789	0.994	0.9826	0.9865	0.9230	0.993	0.9908	0.9844	0.9920
Translate (10, 20)	0.992	0.9912	0.9895	0.9751	0.991	0.9816	0.9843	0.9208	0.993	0.9893	0.9832	0.9897
Translate (20, 35)	0.989	0.9750	0.9859	0.9730	0.989	0.9799	0.9788	0.9208	0.992	0.9890	0.9792	0.9905
Translate (40, 40)	0.983	0.9623	0.9805	0.9657	0.986	0.9752	0.9757	0.9092	0.991	0.9875	0.9732	0.9891

Table 2
Comparative analysis of our scheme with the pervious schemes.

Description	Ganic et al. [11]	Lagzian et al. [20]	Rastegar et al. [9]	Lai et al. [7]	Proposed scheme
Type of scheme	Non-blind	Non-blind	Non-blind	Blind	Blind
Type of transforms	DWT + SVD	RDWT + SVD	Radon + DWT + SVD	DWT + SVD	RDWT + SVD
Embedding sub-bands	All	All	All and LH, HL	$\frac{1}{2}$LH and $\frac{1}{2}$HL	All
Watermark action before embedding	SVD	SVD	SVD	No action	No action
Size of host image	512×512	512×512	257×257	512×512	512×512
Size of watermark	256×256	512×512	33×33	256×256	512×512
Tested host image	Lena	Lena	Lena, Pepper and Baboon	Lena	Lena, Pepper and Baboon
Type of watermark	Gray	Gray	Binary	Gray	Gray
False positive problem	Yes	Yes	Yes	No	No
Scaling factor	0.05 (LL), 0.005 (remaining)	0.05 (LL), 0.005 (remaining)	All (0.05)	All (0.01:0.09)	0.05 (LL), 0.005 (remaining)

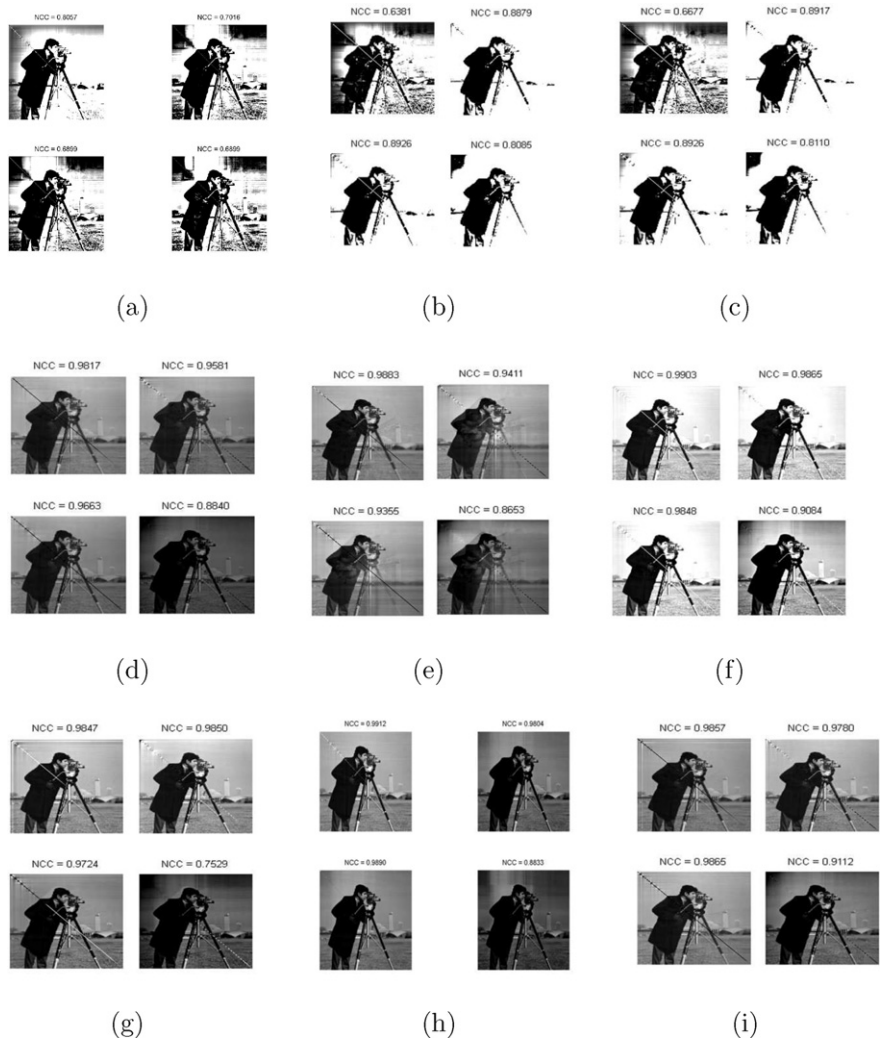

Fig. 8. (a) Extracted watermark after Speckle noise (0.4) (*PSNR* 10.6832 db); (b) extracted watermark after Gaussian noise (0, 0.005) (*PSNR* 23.006 db); (c) extracted watermark after Pepper & salt noise (0.001) (*PSNR* 35.5595 db); (d) extracted watermark after median filter (3 × 3) (*PSNR* 35.4997 db); (e) extracted watermark after JPEG compression *Q* = 40 (*PSNR* 35.0824 db); (f) extracted watermark after HE (*PSNR* 19.1347 db); (g) extracted watermark after rotation (angle 50°) (*PSNR* 10.36672 db); (h) extracted watermark after shift 2% (*PSNR* 15.613 db); (i) extracted watermark after cut attack (*PSNR* 15.1333 db).

5. Experimental results

The proposed scheme is implemented using MATLAB and tested on three standard images, which are considered as the cover images. They are gray scale images of size 512 × 512, identified as: Baboon, Lena and Peppers. The watermark image is a gray-scale image of size 512 × 512. It is identified as Cameraman. These images are shown in Fig. 5. Two different scaling factors are used in the experiments. For LL sub-band, 0.05 is used as a scaling factor while 0.005 for the other three sub-bands. The proposed RDWT–SVD scheme performance is investigated with various experiments in terms of imperceptibility and robustness against various attacks. The imperceptibility means that the human visual quality of the host image should not be affected much even after watermark embedding. Since, the similarity between the host image and the watermarked image must be almost the same. There are objective and subjective criteria to measure the quality of the watermarked image, and both should be good [18]. One of the objective criteria is peak signal-to-noise ratio (*PSNR*), which was employed in this paper to denote the imperceptibility degree.

The *PSNR* can be calculated as follows:

$$PSNR = 10\,log_{10}\left[\frac{max(x(i,j))^2}{MSE}\right] \tag{19}$$

where the Mean Square Error (*MSE*) between the host image *x* and the watermarked image *y* is defined as:

$$MSE = \frac{1}{m * n}\sum_{i=1}^{m}\sum_{j=1}^{n}[x(i,j) - y(i,j)]^2 \tag{20}$$

The subjective criteria can be easily evaluated visually by comparing each cover image; for example as was used in our proposed scheme (Peppers, Lena and Baboon) and shown in Fig. 5. Their corresponding watermarked images are shown in Fig. 6. Fig. 6 is considered as the imperceptibility results of the proposed scheme, when compared to the cover images in Fig. 5. It shows a big similarity between the cover images and their corresponding watermarked images. The second important requirement for any watermarking system is the robustness. The robustness can be evaluated by calculating the Normalized Cross-Correlation (*NC*),which

Table 3
The comparison of robustness for our scheme, Ganic et al. [11] and Lagzian et al. [20].

Attack	Lena image (this paper)				Ganic et al. [11]				Lagzian et al. [20]			
	LL	LH	HL	HH	LL	LH	HL	HH	LL	LH	HL	HH
Pepper & salt noise (density 0.001)	0.994	0.9618	0.9689	0.8727	–	–	–	–	0.9860	0.6246	0.6038	0.6057
Gaussian noise ($M = 0$, var = 0.3)	0.6492	0.8890	0.8927	0.8102	0.865	0.207	0.271	0.277	–	–	–	–
Gaussian noise ($M = 0$, var = 0.005)	0.925	0.8617	0.8156	0.8230	–	–	–	–	0.6774	0.6388	0.6376	0.5836
Speckle noise (var = 0.4)	0.704	0.8908	0.8980	0.8122	–	–	–	–	0.3281	0.6648	0.6838	0.6064
Rotation (angle 50°)	0.985	0.9850	0.9724	0.7529	–	–	–	–	0.5617	0.6803	−0.8282	0.0773
Rotation (angle 20°)	0.9805	0.9755	0.9742	0.9007	−0.353	−0.003	0.963	−0.335	–	–	–	–
JPEG compression Q = 40	0.988	0.9411	0.9355	0.8653	–	–	–	–	0.9724	−0.2217	−0.4009	−0.4247
Median filtering (3 × 3)	0.982	0.9581	0.9663	0.8840	–	–	–	–	0.7218	−0.6752	−0.7912	−0.7564
Shift 2%	0.994	0.9826	0.9865	0.9230	–	–	–	–	0.9999	0.9984	0.9988	0.9907
Cut 20	0.989	0.9820	0.9900	0.9137	–	–	–	–	−0.8881	0.7097	0.0087	0.8292
Histogram Equalization	0.990	0.9865	0.9848	0.9084	0.586	0.657	0.716	0.823	0.5670	0.8138	0.8815	0.8220
JPEG compression Q = 30	0.982	0.8562	0.8753	0.8822	0.993	0.003	0.141	−0.381	–	–	–	–
Sharpening 80	0.9255	0.9314	0.9244	0.9067	0.528	0.553	0.631	0.699	–	–	–	–
Scaling (zoomout = 0.5, zoomin = 2)	0.949	0.6719	0.7038	0.8941	0.940	−0.258	−0.211	−0.437	–	–	–	–
Gamma correction 0.6	0.9894	0.9927	0.9927	0.9263	−0.942	0.946	0.987	0.997	–	–	–	–

indicates the similarity between the original watermark and the extracted watermark after attack. The *NC* is defined as:

$$NC(w,\overline{w}) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[w(i,j)-\mu_w][\overline{w}(i,j)-\mu_{\overline{w}}]}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[w(i,j)-\mu_w]^2}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[\overline{w}(i,j)-\mu_{\overline{w}}]^2}} \tag{21}$$

where N and M represent the number of pixels in the watermark, $w, \overline{w}$ to indicate to the original watermark and the extracted watermark, μ_w and $\mu_{\overline{w}}$ to indicate to the mean of the original watermark and the mean of the extracted watermark respectively. The correlation coefficient between w and $\overline{w}$, can be between −1 and 1. if the NC value is near +1, then the extracted watermark is strongly correlated. If it is near −1, the extracted watermark is also strongly correlated but it looks similar to negative film. But, if it is near 0, the extracted watermark is totally uncorrelated [10]. Generally, the NC is considered acceptable if it is 0.75 or above [19]. Our proposed scheme showed high imperceptibility. This is shown in Fig. 6, where higher *PSNR* values are achieved with all testing images (Pepper image (54.1556 db), Lena image (54.0353 db), and Baboon image (55.9768 db)).

In addition to the imperceptibility, our scheme achieved high robustness. This can be proven by extracting the watermark image and calculating the *NC* between the original watermark and the extracted watermark. Moreover, the watermarked image may be exposed to some attacks during transmission. To make sure the robustness of the proposed scheme, several attacks were applied to all test images. These attacks are image processing attacks such as noise addition attacks, de-noising attacks and compression attacks, and geometrical attacks such as rotation, scaling, shear, cut and translation. Fig. 7 shows Lena host image after being subjected to some attacks while Fig. 8 shows the extracted watermark from the attacked image. Three different types of noise with various density and magnitude were added to the watermarked image in our experiments. There are Salt & pepper noise, Speckle noise and Gaussian noise. Fig. 7(a), (b) and (c) show examples of noising attacks to the watermarked images. The second type of attacks are the filtering attacks. Four types of filters with size 3 × 3 and 5 × 5 window were applied to the watermarked image. These are Median filter, Wiener filter, Gaussian filter and un-sharp filter. Fig. 7(d) shows the watermarked Lena image exposed to 3 × 3 window median filter. Also, different JPEG compression rates were applied to the watermarked image to investigate the scheme robustness. Fig. 7(e) shows 40:1 compressed watermarked image. The remaining image processing attacks were the histogram equalization attack and the gamma correction attack. The histogram equalization attack is shown in Fig. 7(f). Other type of attacks are the geometrical attacks. Five different geometrical attacks were tested using the proposed method. Six different angles of rotation attacks were applied to the watermarked image that are 45°, 2°,70°,110°, −50° and 50°. The 50° rotated watermarked image can be shown in Fig. 7(g). The second type of the geometrical attacks is the scaling attack. In the scaling attack, the watermarked image was resized to different size either smaller or bigger than the original size, then resized again to its original size. The remaining geometrical attacks are cut, shear and translate attacks. Table 1 shows the robustness results of the proposed scheme under various types of attacks. It summarizes the normalized correlation (*NC*) of different host image. Those are Pepper, Lena and Baboon images.

Table 4
The comparison of imperceptibility (db) for our scheme, Lai et al. [7], Rastegar et al. [9] and Lagzian et al. [20].

Image name	Proposed scheme	Lai et al. [7]	Rastegar et al. [9][a]	Rastegar et al. [9][b]	Lagzian et al. [20]
Lena	**54.0353**	50.89	43.3506	45.9337	38.52
Peppers	**54.1556**	–	43.3586	45.9543	–
Baboon	**55.9768**	–	43.3653	45.9228	–

The bold values indicates the best values comparing with the others.
[a] Indicates the first scheme in Rastegar et al. [9]
[b] Indicates the second scheme in Rastegar et al.[9]

Table 5
The comparison of robustness for our scheme, Lai et al. [7] and Rastegar et al. [9]. '–' means the attacks are not done.

Attack	Proposed scheme	Lai et al. [7]	Rastegar et al. [9][a]	Rastegar et al. [9][b]
Pepper & salt noise (density 0.3)	**0.8926**	–	0.7515	0.8258
Speckle noise (var = 0.01)	0.952	–	0.9609	**0.9667**
Gaussian noise (M = 0, var = 0.5)	**0.8935**	–	0.7926	0.82
Gaussian filtering (3×3)	**0.987**	–	0.8023	0.9843
Median filtering (3×3)	**0.982**	0.9597	0.7534	0.9706
Wiener filtering (3×3)	**0.984**	–	0.9824	0.9569
Sharpening	0.932	–	**0.9687**	0.9511
Histogram equalization	**0.990**	0.9862	0.9648	0.9628
Gamma correction (0.7)	0.9935	**0.9982**	–	–
Gamma correction (0.8)	**0.9950**	–	0.7203	0.9217
JPEG compression Q = 30	**0.987**	–	–	–
JPEG compression Q = 10	0.972	**0.9772**	0.9824	**0.9843**
JPEG compression Q = 5	**0.952**	–	0.8532	0.9354
Scaling (zoomout = 0.5, zoomin = 2)	0.948	–	0.5127	**0.953**
Rotation (angle = 2°)	**0.981**	–	0.5068	0.9628
Rotation (angle = −30°)	**0.9823**	0.9780	–	–

The bold values indicates the best values comparing with the others.
[a] Indicates the first scheme in Rastegar et al. [9]
[b] Indicates the second scheme in Rastegar et al.[9]

6. Comparative analysis and discussion

6.1. Comparative analysis

In this section, the proposed scheme is investigated by comparing its results with Lagzian's [8], Ganic's [11], Lai's [7], and Rastegar's [9] schemes. The similarities, differences, and some of the highlighted points in each scheme are summarized in Table 2.

As shown in Table 2, Ganic's, Lagzian's and Rastegar's schemes applied SVD on the watermark image before the embedding step. This is the main weakness in their systems. This may cause a false-positive problem [21,10]. Lai et al. scheme solved this problem, but lesser information could be embedded in their scheme. They divided the 256×256 gray image watermark into two parts and embedded one part in LH sub-band and the other in HL sub-band. In our scheme, a 512×512 gray image was embedded in each sub-band. Rastegar et al. [9] applied radon transform before DWT and SVD. They embedded a 33×33 binary image into the sub-bands of 257×257 host images. According to the differences and similarities between all schemes, a comparison between their results can be done. Our scheme results are compared with Lagazian's and Ganic by comparing each sub-band's result. The *NC* values of the

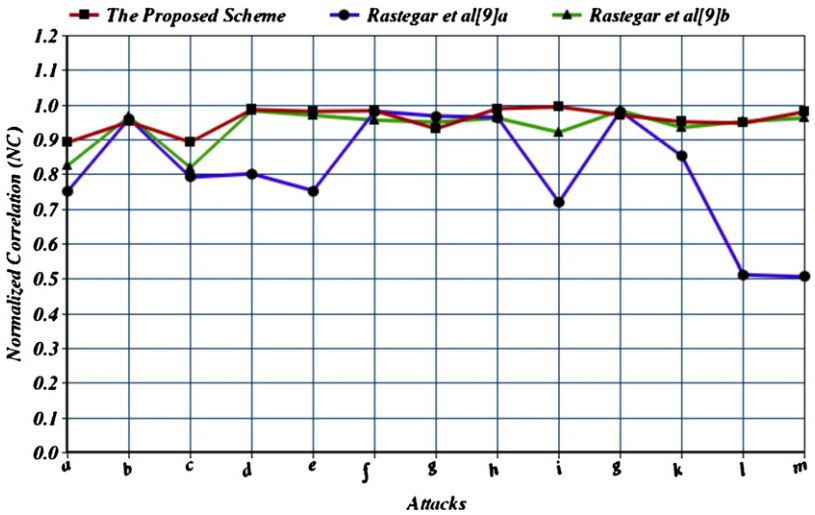

Fig. 9. Robustness comparison of the proposed scheme with Restagar's schemes. (a) Pepper & salt (density 0.3); (b) Speckle noise (0.01); (c) Gaussian noise (0, 0.3); (d) Gaussian filter (3×3); (e) median filter (3×3); (f) Wiener filter (3×3); (g) sharpening; (h) histogram equalization; (i) gamma correction 0.8; (j) JPEG compression Q = 10%; (k) JPEG compression Q = 5%; (l) scaling (zoomout = 0.5, zoomin = 2); (m) rotation (angle = 2°).

Table 6
The *NC* of the extracted watermark under different attacks.

Attack	DWT-SVD				RDWT–SVD			
	LL	LH	HL	HH	LL	LH	HL	HH
JPEG compression $Q=5$	**0.9767**	**0.8402**	**0.8884**	**0.8573**	0.952	0.7666	0.7190	0.7487
JPEG compression $Q=10$	**0.9882**	**0.9261**	**0.9524**	**0.8795**	0.972	0.8557	0.8328	0.7901
Pepper & salt noise (density 0.3)	0.6406	**0.8944**	0.8909	**0.8446**	**0.668**	0.8917	**0.8926**	0.8110
Speckle noise (var = 0.4)	0.6897	0.8882	0.8955	**0.8386**	**0.704**	**0.8908**	**0.8980**	0.8122
Gaussian noise ($M=0$, var = 0.005)	**0.9431**	**0.8798**	0.7906	0.8059	0.925	0.8617	**0.8156**	**0.8230**
Histogram equalization	0.9852	0.9854	0.9845	**0.9825**	**0.990**	**0.9865**	**0.9848**	0.9084
Median filtering (5×5)	**0.9582**	**0.9309**	**0.9378**	0.9811	0.950	0.9130	0.9268	**0.9845**
Wiener filtering (5×5)	**0.9771**	**0.9690**	**0.9682**	**0.9711**	0.974	0.9404	0.9561	0.9427
Gaussian filtering (5×5)	0.9885	0.9317	**0.9744**	**0.9828**	**0.987**	**0.9649**	0.9606	0.9418
Sharpening	0.9230	**0.9342**	**0.9320**	0.9142	**0.927**	0.9312	0.9237	0.9094
Scaling (zoomout = 0.5, zoomin = 2)	**0.9778**	**0.9001**	**0.8713**	**0.9378**	0.948	0.8094	0.7467	0.8605
Scaling (zoomout = 0.25, zoomin = 4)	**0.8471**	**0.6537**	**0.5926**	**0.9065**	0.788	0.5588	0.5137	0.8528
Rotation (angle 45°)	0.9684	0.9763	0.8920	**0.8924**	**0.983**	**0.9823**	**0.9815**	0.8045
Rotation (angle 70°)	0.9721	**0.9690**	0.8653	0.8507	**0.981**	0.9668	**0.9436**	**0.8619**
Rotation (angle −50°)	0.9695	0.9726	**0.9493**	**0.8071**	**0.984**	**0.9751**	0.9376	0.7810
Shift 2%	**0.9995**	**0.9994**	**0.9994**	**0.9989**	0.9912	0.9804	0.9890	0.8833
Shearing ($x=1$, $y=0.2$)	0.9622	**0.9769**	0.9744	0.8943	**0.980**	0.9554	**0.9862**	0.8752
Shearing ($x=0.2$, $y=1$)	0.9229	**0.8963**	0.7434	0.6886	**0.957**	0.8621	**0.7274**	**0.7813**
Cut 20	0.9846	0.6528	0.9846	**0.9903**	**0.989**	**0.9820**	**0.9900**	0.9137
Cut 30	0.9811	0.6305	0.9820	**0.9888**	**0.986**	**0.9780**	**0.9865**	0.9112
Translate (20, 35)	0.9823	**0.9874**	0.9719	**0.9343**	**0.989**	0.9799	**0.9788**	0.9208
Translate (40, 40)	0.9755	**0.9879**	0.9719	**0.9899**	**0.986**	0.9752	**0.9757**	0.9092

The bold values indicates the best values comparing with the others.

proposed scheme, Lagzian [8] and Ganic [11] schemes are shown in Table 3. From Table 3, we can summarize the following:

1. Our proposed scheme is more robust than Lagzian and Ganic schemes.
2. Unlike the previous schemes [8,11], the proposed scheme has consistent response across all sub-bands.
3. Our scheme achieved better resistance against the geometrical attacks.
4. LL sub-band had the better response against almost of the attacks except in the case of a high percentage of noise addition.

Besides the robustness results, the proposed scheme achieved high imperceptibility compared with Lagzian [8] and other schemes. This is shown in Table 4. The results showed that our RDWT–SVD method is more robust than the Lagzian's method. All sub-bands showed good robustness against all attacks. In Lagzian's method for some attacks, the LL sub-band showed robustness against some attacks but the others sub-bands failed to resist against the same attacks. For example for Lagzian, as shown in Table 3 for JPEG compression attack; the NC of extracted watermark from LL sub-band was 0.9724. However, it was −0.2217, −0.4009, and −0.4247 for LH, HL, and HH sub-bands respectively.

In order to compare with the remaining schemes, the highest *NC* values in our scheme for each attack were selected for comparison. Table 5 shows the proposed results with Rastegar's schemes results and Lai's scheme results when scaling factor is 0.05. Rastegar scheme (a) represents the embedding in all sub-bands while Rastegar scheme (b) represents the embedding in LH and HL only. Our scheme performed better than Lai's and Rastegar's schemes as shown in Table 5. The robustness of the proposed scheme and both Rastegar's schemes are summarized in Fig. 9.

Although, the proposed scheme compared with some of DWT-SVD schemes with different embedding algorithm, we compared also the same embedding scheme using DWT-SVD. The comparison is shown in Table 6. Both schemes showed robustness but the superiority and strength of our proposed scheme RDWT–SVD lies on the amount of embedded information. It was double the information that can be embedded by DWT for each sub-band. In addition, our proposed scheme showed more robustness against almost of the geometrical attacks.

6.2. Conclusion

The redundancy in the RDWT domain portrayed better robustness in carrying watermarking information where the watermark was directly embedded to all the RDWT sub-bands of the host image. Although more embedded data may lead to image distortion, the proposed scheme preserved high image imperceptibility. This was to the fact that SVD has good stability. Furthermore, the LL sub-band appeared to be more robust than other sub-bands for almost all attacks except the noise attacks. In the case of a large amount of noise, the LH and HL were more robust. The experimental results demonstrated that the proposed scheme achieved a high robustness against geometrical and non-geometrical attacks. It represented a good choice for copyright protection because of the blind issue. This is because the watermark was embedded directly to the cover image without applying SVD on it. That is to solve the false-positive problem by avoiding using the watermark's SVD orthogonal vectors U and V in the embedding and extraction process. Thus, our proposed scheme has satisfied the capacity, robustness, imperceptibility, and security requirements.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions. This work is supported by Ministry of Higher Education, Malaysia through Fundamental Research Grant Scheme (203/PELECT/6071135).

References

[1] Hien TD, Nakao Z, Chen YW. Robust multi-logo watermarking by RDWT and ICA. Signal Process 2006;86(10):2981–93.
[2] Li Z, Ping-ping Z, Gong-bin Q, Zhen J. Image watermarking with optimum capacity. In: 5th international conference on visual information engineering. 2008. p. 117–23.
[3] Wu HT, Cheung YM. Reversible watermarking by modulation and security enhancement. IEEE Trans Instrum Meas 2010;59(1):221–8.
[4] Campisi P, Kundur D, Neri A. Robust digital watermarking in the ridgelet domain. IEEE Signal Process Lett 2004;11(10):826–30.

[5] Lin S, Chen CF. A robust DCT-based watermarking for copyright protection. IEEE Trans Consum Electron 2000;46(3):415–21.
[6] Premaratne P, Ko C. A novel watermark embedding and detection scheme for images in DFT domain. In: 7th international conference on image processing and its applications. 1999. p. 780–3.
[7] Lai CC, Tsai CC. Digital image watermarking using discrete wavelet transform and singular value decomposition. IEEE Trans Instrum Meas 2010;59(11):3060–3.
[8] Lagzian S, Soryani M, Fathy M. A new robust watermarking scheme based on RDWT–SVD. IJIIP Int J Intell Inform Process 2011;2(1):22–9.
[9] Rastegar S, Namazi F, Yaghmaie K, Aliabadian A. Hybrid watermarking algorithm based on singular value decomposition and radon transform. Int J Electron Commun (AEU) 2011;65(7):658–63.
[10] Bhatnagar G. A new facet in robust digital watermarking framework. Int J Electron Commun (AEU) 2012;66(4):275–85.
[11] Ganic E, Eskicioglu AM. Robust embedding of visual watermarks using discrete wavelet transform and singular value decomposition. J Electron Imaging 2005;14(4):043004–9.
[12] Lai CC. An improved SVD-based watermarking scheme using human visual characteristics. Opt Commun 2011;284(4):938–44.
[13] Ling HC, Phan RCW, Heng SH. On the security of a hybrid watermarking algorithm based on singular value decomposition and radon transform. Int J Electron Commun (AEU) 2011;65(11):958–60.
[14] Zhang XP, Li K. Comments on "an SVD-based watermarking scheme for protecting rightful ownership". IEEE Trans Multimedia 2005;7(3): 593–4.
[15] Fowler J. The redundant discrete wavelet transform and additive noise. IEEE Signal Process Lett 2005;12(9):629–32.
[16] Cui S, Wang Y. Redundant wavelet transform in video signal processing. In: International conference on image processing, computer vision, pattern recognition. 2006. p. 191–6.
[17] Haque SR. Singular value decomposition and discrete cosine transform based image watermarking. Master's thesis. Sweden: Department of Interaction and System Design, School of Engineering, Blekinge Institute of Technology; 2008.
[18] Cox I, Miller M, Bloom J, Fridrich J, Kalker T. Digital watermarking and steganography. 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2007.
[19] Al-Haj A. Combined DWT-DCT digital image watermarking. J Comput Sci 2007;3(11):740–6.
[20] Lagzian S, Soryani M, Fathy M. Robust watermarking scheme based on RDWT–SVD: embedding data in all subbands. In: International symposium on artificial intelligence and signal processing (AISP). 2011. p. 48–52.
[21] Mohammad A, Alhaj A, Shaltaf S. An improved SVD-based watermarking scheme for protecting rightful ownership. Signal Process 2008;88(9): 2158–80.