UNIVERSITY OF TARTU

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Institute of Computer Science

Software Engineering Curriculum

Lauri Laur

Entropy Based Robust Watermarking Algorithm

Master's Thesis (30 ECTS)

Supervisors: Assoc. Prof. Gholamreza Anbarjafari

Asst. Prof. Mary Agoyi

Tartu 2015

Entropy Based Robust Watermarking Algorithm

Abstract

With growth of digital media distributed over the Internet, concerns about security and piracy have emerged. The amount of digital media reproduction and tampering has brought a need for content watermarking. In this work, multiple robust watermarking algorithms are introduced. They embed watermark image into singular values of host image's blocks with low entropy values. In proposed algorithms, host image is divided into blocks, and the entropy of each block is calculated. The average of all entropies indicates the chosen threshold value for selecting the blocks in which watermark image should be embedded. All blocks with entropy lower than the calculated threshold are decomposed into frequency subbands using discrete wavelet transform (DWT). Subsequently chirp z-transform (CZT) is applied to the low-frequency subband followed by an appropriate matrix decomposition such as lower and upper decomposition (LUD) or orthogonal-triangular decomposition (QR decomposition). By applying singular value decomposition (SVD) to diagonal matrices obtained by the aforementioned matrix decompositions, the singular values of each block are calculated. Watermark image is embedded by adding singular values of the watermark image to singular values of the low entropy blocks. Proposed algorithms are tested on many host and watermark images, and they are compared with conventional and other state-of-the-art watermarking techniques. The quantitative and qualitative experimental results are indicating that the proposed algorithms are imperceptible and robust against many signal processing attacks.

Keywords:

Chirp Z Transform, Discrete Wavelet Transform, Entropy, Image Security, Lower and Upper Decomposition, Orthogonal-Triangular Decomposition, Singular Value Decomposition, Watermarking

Entroopial Põhinev Vastupidav Vesimärgi Algoritm

Lühikokkuvõte

Tänu aina kasvavale multimeedia andmeedastus mahtudele Internetis, on esile kerkinud mured turvalisusest ja piraatlusest. Digitaalse meedia paljundamise ja muutmise maht on loonud vajaduse digitaalse meedia vesimärgistamise järgi. Selles töös on tutvustatud vastupidavaid vesimärkide lisamise algoritme, mis lisavad vesimärgid madala entroopiaga pildi osadesse. Välja pakutud algoritmides jagatakse algne pilt blokkidesse ning arvutatakse iga bloki entroopia. Kõikide blokkide keskmine entroopia väärtus valitakse künniseks, mille järgi otsustatakse, millistesse blokkidesse vesimärk lisada. Kõik blokid, mille entroopia on väiksem kui künnis, viiakse signaali sageduse kujule kasutades *Discrete Wavelet Transform* algoritmi. Madala sagedusega sagedusvahemikule rakendatakse *Chirp Z-Transform* algoritmi ja saadud tulemusele LU-dekompositsiooni või QR-dekompositsiooni. *Singular Value Decomposition* meetodi rakendamisel diagonaalmaatriksile, mis saadi eelmisest sammust, saadakse iga bloki vastav väärtus. Vesimärk lisatakse pildile, liites iga bloki arvutatud väärtusele vesimärgi *Singular Value Decomposition* meetodi tulemused. Kirjeldatud algoritme testiti ning võrreldi teiste tavapärast ning uudsete vesimärkide lisamise tehnoloogiatega. Kvantitatiivsed ja kvalitatiivsed eksperimendid näitavad, et välja pakutud meetodid on tajumatud ning vastupidavad signaali töötlemise rünnakutele.

Võtmesõnad:

Chirp Z Transform, Discrete Wavelet Transform, Entroopia, LU-dekompositsioon, Pildi turvalisus, QR-dekompositsioon, *Singular Value Decomposition*, Vesimärk

Content

1. Introduction

1.1. Cyber security and privacy

The Internet was opened for commercial use in the early 1990s. Nowadays the Internet plays a fundamental role in our everyday life, it supports basic information infrastructure. Originally it was meant for information-sharing, hence security was not a big part of it [1]. Cyber security aims to guard networks, computers, programs and data from unauthorized access, change or destruction. Together with the amount of data shared over the Internet, questions about how to protect the data arise. Digital media content can be easily distributed, processed, duplicated and modified. Because of this, it is necessary to implement systems that would maintain copyright, protect the integrity and do copyright control of digital media [2].

Privacy can be defined as a right to be let alone. Privacy invasion occurs when an individual cannot maintain control over its personal information usage [3]. Privacy protection is an emerging field. Illegal distribution, plagiarism and copyright violation represent a threat to content producers and owners, because it makes more difficult to sell their content with profitable price [4].

1.2. Watermarking

Digital data hiding has received increasing attention from information technology community from 1990's. With popularization on the World Wide Web and ease of data transferring over the Internet, copyright and security concerns have emerged. As computational performance has become cheaper it is nowadays cheaper to copy and distribute digital media than make one. In order to protect digital media from piracy, unauthorized use and other illegal actions, watermarks are used. Most of the demand for watermarking comes from movie, music and picture industries, where piracy is a big concern.

Watermarking is a method used in computer security where identifiers of the signal holder are embedded in the host signal for keeping track where the signal comes or who are the copyright owners. Signal carrying the information before watermark is embedded into is called cover signal or host and the data holding copyright ID is referred as watermark [5]. When some digital media file is received, watermark can be extracted from it and it can provide authentication to digital media and protect the copyright [6].

Watermarking algorithms can be blind, semi-blind and non-blind algorithms. Method is blind when original objects are not used to extract the watermark from watermarked signal and only secret key is required. Semi-blind watermarking techniques need a secret key and the original watermark to extract embedded watermark from watermarked signal. Non-blind watermarking technique requires original signal, watermark sequence and the secret key to extract the watermark from watermarked image [7]. Non-blind algorithms are usually more robust than blind ones, because when data to which watermark is embedded is unknown, it is treated as disturbing noise [8]. Original signal of watermarked content may not always be available in many applications, therefore blind methods will be implemented. Non-blind techniques are appropriate for some applications, for example, when the owner of the signal needs to prove ownership [9].

Watermarking algorithm has many requirements for properties that it has to fulfil. It is hard to fulfil all requirements at the same level. It is likely that when one property is very strong it comes from another property's expense. For example, when a watermarking algorithm is very robust it is likely that imperceptibility will not be the highest. When making good watermarking algorithm author has to accept trade-off between properties.

1.3. Watermarking history

Secret communication is as old as communication itself. Steganography comes from old Greek and it means protected writing. Nowadays steganography is known as embedding message inside another message. The main difference between steganography and watermarking is that watermarking requires the watermark to be robust against attacks. 700 years ago appeared paper watermarks in handmade papers. In the digital data context, watermarking was inspired by watermarks in money bills and in stamps. Digital image watermarking started in 1990 when K. Tanaka, Y. Nakamura and K. Matsui proposed a method for embedding secret information into digital image [10]. Starting from 1995 watermarking received great attention, it has evolved quickly since then and there are still many topics that need more research regarding this field [11]. In 1996 first Information Hiding Workshop [12] was held that included digital watermarking as one of its primary topics [13]. Many techniques used in steganography in the past have reappeared in modern data-embedding and watermarking literature [14].

1.4. Non-blind watermarking application areas

Non-blind watermarking algorithms have many advantages over blind watermarking algorithms and they can be used in many different application areas. Copyright watermarking helps to ensure that owner's or producer's identification stays permanently attached to the content. Fingerprint watermarking assures copyright protection while customer's data is embedded into digital media to track legal and illegal copies [15]. Broadcast monitoring helps advertisers to verify that they will only pay for those commercials that were really broadcasted, by embedding watermark into advertisements [16]. In authentication watermarking helps to identify any forgery or tampering of the original content if the watermark is missing. This is crucial in legal cases and in medical images [17].

1.5. Proposed algorithms

Multiple non-blind watermarking algorithms are going to be proposed in this work. The algorithms are going to be non-blind, because typically non-blind watermarking algorithms are more robust against attacks. Experimental results are conducted and these are used for comparison between proposed algorithms and other already published techniques. Proposed methods use entropy to evaluate to which parts of the signal watermark is going to be embedded. These methods take advantage of several mathematical tools such as singular value decomposition (SVD) and lower upper triangular matrix decomposition (LUD) and signal processing techniques like discrete wavelet transform (DWT) and chirp z-transform (CZT), combining them to develop robust and imperceptible the state-of-the-art watermarking schemes. Robustness of proposed algorithms are measured after several signal processing

attacks like cropping, flipping, histogram equalization, and JPEG conversion are applied to watermarked image. Such watermarking schemes can be used in many security related areas like forensics, ownership proving, corporate document and image security and many others.

1.6. Thesis structure

The remaining parts of the thesis are structured as follows: second chapter gives an overview about watermarking background, its properties and commonly used methods. The third chapter presents other developed watermarking techniques and their properties. In the fourth chapter, detailed description together with flowcharts, equations and pseudocode about proposed methods is given. Chapter five presents the experimental results and the outcome. Chapter six concludes the thesis and brings out possible future work opportunities.

2. Background

2.1. Watermark properties

2.1.1. Robustness

Robustness is very important because it is needed for detecting watermark after common signal processing attacks. Removal or tampering of watermark may be intentional or unintentional and it can be done by simple image processing attacks like blurring, flipping, contrast enhancement, gamma correction, and noise adding [18]. In fact, in order to be robust, the watermarking pattern should be embedded with high power, but this power implies a substantial distortion of the original media content.

2.1.2. Imperceptibility

Imperceptibility basically means that there should be no perceptible difference between the original and the watermarked signal. When adding watermark to digital image, it should not affect the visual quality of the original image. Watermarks imperceptibility can be expressed as a metric between watermarked and original image [18, 19]. Usually if any perceptual distortions are introduced, it reduces the commercial value of the content.

2.1.3. Security

Security requirement states that it should not be easy to remove or change the watermark without changing the host signal. The threats that face watermarking algorithm depend where this algorithm is going to be used. Some applications require bigger security, others may not. For example watermarking used in authentication of legal documents and in medical images require a bigger level of security [20].

2.1.4. Capacity

Capacity restriction refers to the constraint that how much information can be embedded in the host signal without damaging it. Watermark should contain at least the minimum amount of information what is needed for representing the uniqueness of signal. Capacity of watermark depends mainly on the content of the signal and the strength of watermark and it is constrained by robustness, reliability and fidelity [18, 21].

2.2. Watermarking algorithm domains

Watermarking algorithms are generally grouped by domain into two groups, spatial and frequency domain algorithms. Spatial domain algorithms embed watermark into digital image by modifying its pixel values. Most widely known algorithm in spatial domain is least significant bit (LSB) method, where watermark is embedded into host images least significant bit values of every pixel. Spatial domain algorithms are easy to implement and they have low complexity, on the other hand, they are not robust against signal processing attacks. Spatial domain watermarking techniques do not change the quality of an image, they assure a high invisibility [22, 23, 24].

Frequency domain algorithms embed the watermark by modifying the digital media magnitude coefficient according to the embedding algorithm. Frequency domain algorithms have bigger computational cost, but they are more robust against common signal processing attacks. In frequency domain methods watermark image is irregularly spread all over the host image, which makes it difficult for the attacker to modify, decode or read. Frequency domain has many different algorithms, most commonly used ones are discrete wavelet transform, discrete cosine transform (DCT) and discrete Fourier transform (DFT) [22, 23].

2.2.1. Discrete wavelet transform

Discrete wavelet transform is a method for transforming a digital image by decomposing it into a set of frequency channels. DWT is a time-scale representation of digital signal. It is obtained with digital filtering techniques and it is calculated by successive high-pass and low-pass filtering of the discrete time-domain signal. Graphical representation of DWT is presented in Figure 1. There are various filters available. However, the most commonly used are Haar Wavelet Filter, Daubechies Bi-Orthogonal Filters and Daubechies Orthogonal Filters [18]. When the input sequence is constant, decomposition in the Haar basis eliminates high frequency terms, hence Haar function is used when images have high contrast of black and white. Haar filter is a special case of Daubechies filter family, it is Daubechies filter of order 1. Daubechies filter construction is based on calculating the frequency response function for the filter coefficients fulfilling moment and orthogonality conditions. Orthogonality and asymmetry are the main features of Daubechies family [25]. Four bands of data, low-frequency band (LL), vertical mid-frequency band (LH), horizontal mid-frequency band (HL) and high-frequency band (HH) are produced in 1-level 2-dimensional DWT. In n-level 2-dimestional DWT, the LL subband is subject of being decomposed into further subband images by applying DWT n-1 times. Due to multi-resolution characteristics, watermark can be embedded into each of those bands. Generally watermark is added into LL because it is more robust against attacks. Modifying HH band is not reducing imperceptibility so much that human eye can detect it, but the robustness is compromised [26, 27].

DWT has many good characteristics. Input image is decomposed into three spatial directions, namely, horizontal, vertical and diagonal in wavelet transform. That’s why wavelets reflect more precisely anisotropic properties of the human visual system. Using simple filter convolution, wavelet transform can be easily implemented and it is computationally efficient. In the lower resolution, watermark detection is also computationally effective, since there are few frequency bands involved at every successive resolution level. High resolution subbands can be used to detect edges and texture patterns in an image [18].

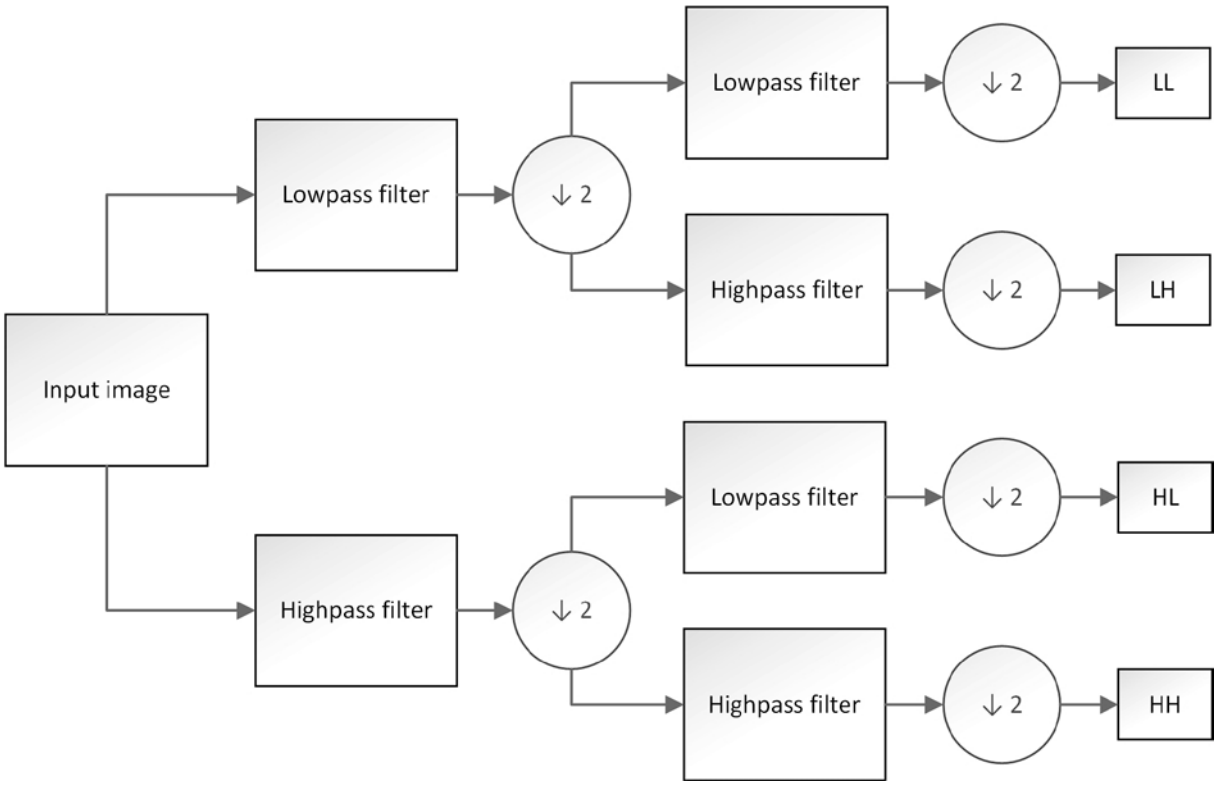


Figure 1. Graphical representation of DWT.

2.2.2. Discrete cosine transform

Discrete cosine transform was defined in 1974 [28]. Like DWT, DCT transforms signal from spatial representation to frequency representation. Rakhi Dubolia *et al.* compared DWT and DCT, and found that using DWT in image watermarking gives better image quality than using DCT [29]. DCT is also used in well know lossy compression Joint Photographic Experts Group (JPEG) [30]. DCT has many advantages like small bit error rate, high compression ratio, good synthetic effect of calculation complexity and good information integration ratio. It also allows the image to be divided into different frequency bands, what makes it easier to embed watermark into middle frequencies of image [31].

2.2.3. Discrete Fourier transform

Discrete Fourier transform is equivalent to the continuous Fourier transform in discrete-world. DFT is widely used to calculate numerically the Fourier transform of functions or signals [32]. It maps time series data from the spatial domain into the frequency domain. DFT is useful because it is rotation, scaling and translation invariant. Strongest components of DFT are central components, which contain low frequencies [18].

2.3. Typical watermarking techniques

2.3.1. Singular value decomposition

Singular value decomposition is a technique in linear algebra to diagonalize matrices. It stores most of the signal's energy in singular values. Singular values of signal are not very sensitive about signal processing attacks and they have typical algebraic image characteristics [33].

SVD is based on a theorem which states that rectangular matrix *A* can be split up into the product of three matrices as shown in Equation 1. *U* is an orthogonal matrix, Σ is a diagonal matrix and *V* is the transpose of an orthogonal matrix [34].

$$A_{m\times n} = U_{m\times m} \cdot \Sigma_{m\times n} \cdot V^{T}_{n\times n}$$

Equation 1

Using SVD in watermarking has many advantages. Singular value of an image is stable. It changes little when the signal is attacked. Singular values of image represent the luminance of an image and the corresponding singular vectors show geometric properties [35]. Matrix sizes from SVD are not fixed and they don't have to be square matrices.

2.3.2. Entropy

Entropy shows how much information signal has. The higher entropy value is, the more information source contains [36]. In image processing, it means that the entropy is great on an image where uncertainty and complexity is large. Also more watermark information can be embedded into host image when host images entropy is large [37]. Entropy can be used in digital media watermarking by embedding watermark into signals low entropy parts. This makes watermarking algorithm more robust against signal processing attacks, because common attacks change dramatically high entropy parts of signal. Entropy *E* can be calculated as shown in Equation 2, where $P(a_j)$ refer to the source symbol/pixel probabilities and *J* refers to the number of symbols or different pixel values [38].

$$E = -\sum_{j=1}^{J} P(a_j) \log_2 \mathrm{P}(a_j)$$

Equation 2

2.3.3. Lower and upper decomposition

Lower and upper decomposition states that any square matrix can be presented as a product of lower and upper triangular matrices by performing a sequence of Gaussian eliminations [39]. Lower triangular matrix has ones in diagonal, multipliers below the diagonal and zeros above the diagonal. Upper triangular matrix has coefficients in the diagonal, multipliers above diagonal and zeros below diagonal [40]. After doing LUD on an image, it is easy to see that upper triangular matrix values are relatively large, and this means that they can be used to embed a watermark [41]. LUD is calculated by solving a series of linear equations of the form in Equation 3, where *A* is non-singular $m \times m$ square matrix and x and b are $m \times 1$ column vectors. The results of such decomposition are lower triangular matrix *L* and an upper triangular matrix *U*, which are shown in Equation 4 [42].

$$Ax = b$$

Equation 3

$$LUx = b$$

Equation 4

3. Conventional and the state-of-the-art techniques in watermarking

3.1. Least significant bit method

Least Significant Bit (LSB) method is one of the simplest watermarking techniques. It operates in spatial domain and modifies image's pixel values. Algorithm for embedding black-and-white image into 8-bit grayscale image is following. Read host image pixel value and modify its least significant bit (8-th bit) to be the same as watermark image's corresponding pixel value [43]. 8-bit image's each pixel's maximum value is 255 which takes maximum of eight bits to represent. While modifying least significant bit, pixel value changes by one, so there is not much visual difference between original pixel and modified pixel.

LSB method is very easy to understand and implement and it requires small computational cost. It is very vulnerable to attacks and if the algorithm is discovered, it is very easy to change or read the hidden information by intruder [44].

3.2. Lai & Tsai method

Chih-Chin Lai and Cheng-Chih Tsai proposed watermarking method [45] using discrete wavelet transform and singular value decomposition. They state that their approach will not need as much computation as other algorithms to compute SVD. Also to preserve better visual perception of the original image they embed watermark to singular values of the cover image instead of embedding singular values of watermark like most existing DWT-SVD-based watermarking algorithms do.

To embed watermark they use Haar DWT on cover image to decompose it into four subbands. Then they use SVD on decomposed vertical and horizontal mid-frequency bands. Afterwards singular values are modified with half of the watermark multiplied with scale factor and SVD is applied to them respectively. Scale factor is used to control the strength of the watermark. After that there are two modified DWT subbands and using these, inverse DWT is applied to get watermarked image. Flowchart of embedding algorithm is presented in Figure 2.

In order to extract watermark from image, DWT is applied to watermarked image. After that SVD is applied on decomposed vertical and horizontal mid-frequency bands. Afterwards half of the watermark is extracted from each subband. Lastly two extracted watermark parts are combined to retrieve embedded watermark image. Extraction algorithm flowchart is shown in Figure 3.

Authors ran several experiments using grayscale images. To measure perceptual quality they used peak signal to noise ratio (PSNR). For robustness measuring, they used Pearson's correlation coefficient. In order to justify their approach they compared proposed method with DWT-SVD based watermarking method [46] and pure SVD-based approach [47]. Experimental results show that proposed algorithm outperforms two compared schemes.

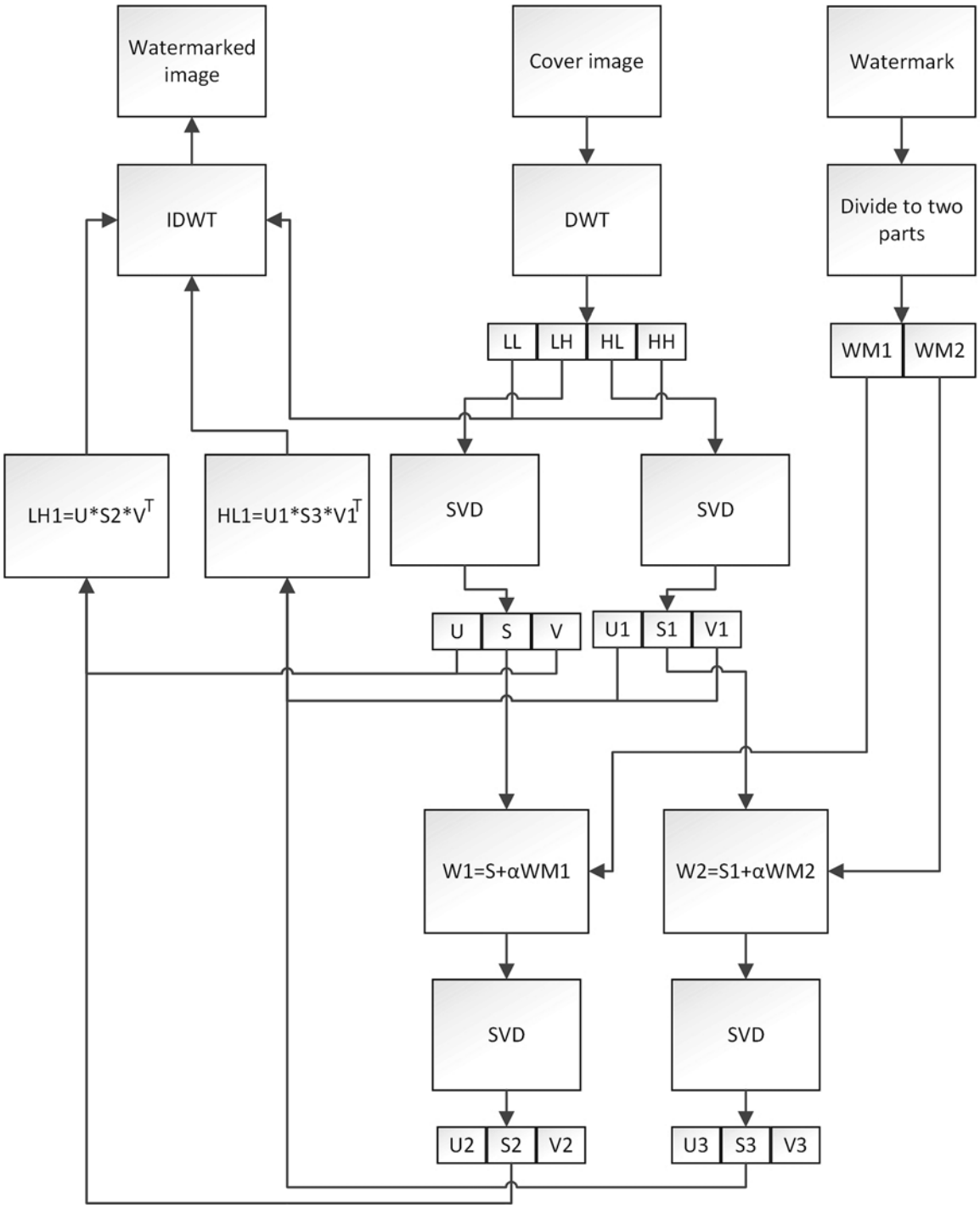


Figure 2. Lai & Tsai method's watermark embedding algorithm flowchart.

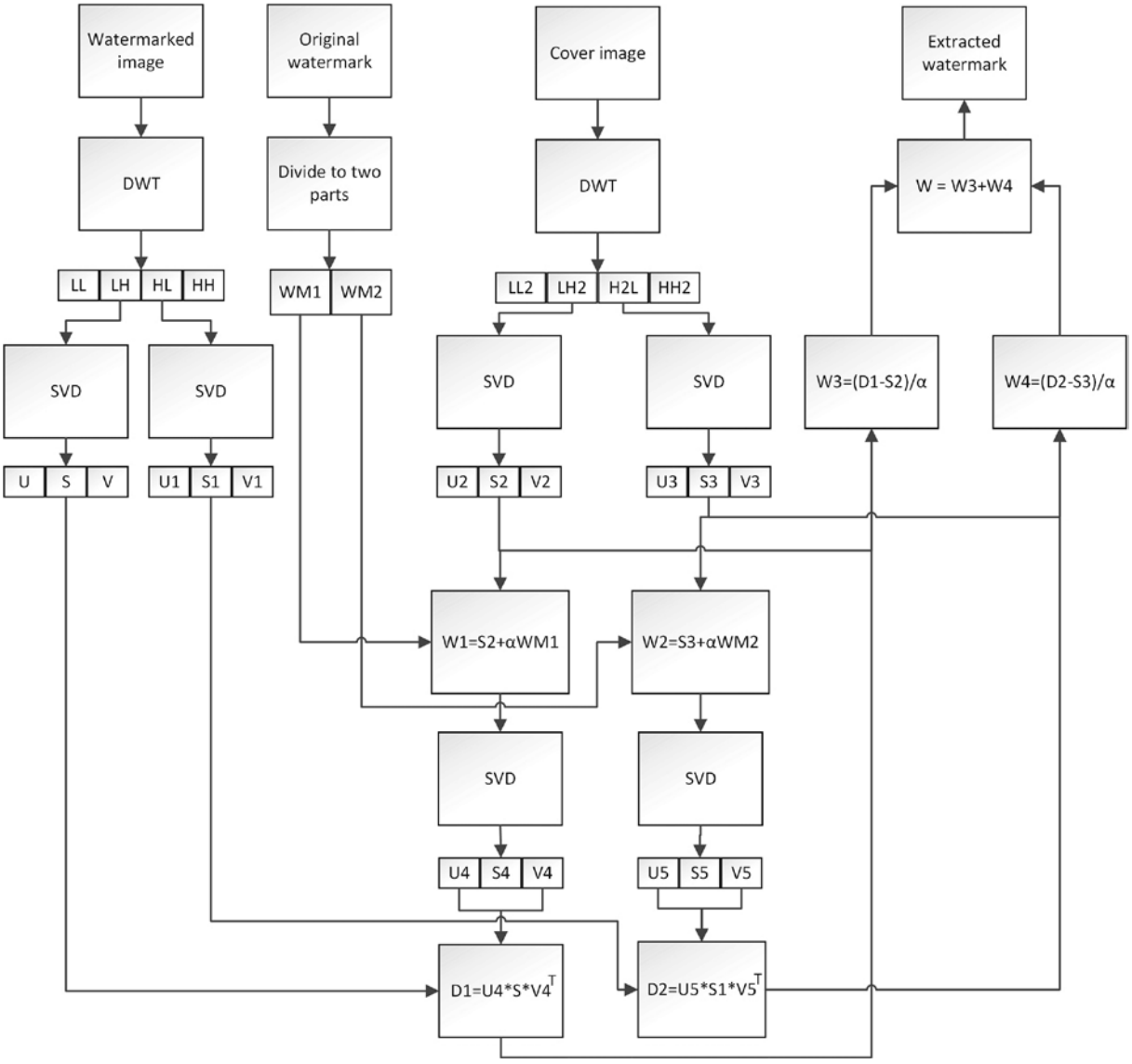


Figure 3. Lai & Tsai method's watermark extraction algorithm flowchart.

3.3. A watermarking algorithm based on chirp z-transform, discrete wavelet transform, and singular value decomposition

Mary Agoyi *et al.* proposed a watermarking algorithm [22] what takes advantages of three widely used technologies in watermarking. It uses discrete wavelet transform in combination with singular value decomposition and chirp z-transform. In their work, they carried out several experiments. For experiments they used four gay scale images as host images and one symmetric and one non-symmetric image as watermark. They compared experimental results with Lai & Tsai method [45] and they found that their algorithm gives better robustness and imperceptibility results. To measure imperceptibility, peak signal to noise ratio (PSNR) and the structural similarity (SSIM) were used. PSNR is measured in decibels and it defines the resemblance between original image and watermarked image. Higher PSNR value means that watermarked image closely resembles the original image. SSIM index assesses similarity between two images. To measure robustness they used SSIM index and correlation coefficient. They calculated quantitative results using given measures and found out that their

method shows superior results over the state-of-the-art algorithm what is proposed in Lai & Tsai method [45].

Watermark embedding algorithm applies DWT to input image to decompose it into four subbands. After that CZT is applied to high-frequency band and further SVD is used on previous step result. Watermark is added by modifying singular value of decomposed image using watermarks singular value and scaling factor. Afterwards orthogonal matrices of host image are combined with modified singular value and inverse CZT is applied to that result. From that, modified high-frequency band is gotten and it is used together with original image's other three subbands to calculate inverse DWT to get watermarked image. Flowchart of watermark embedding is shown in Figure 4.

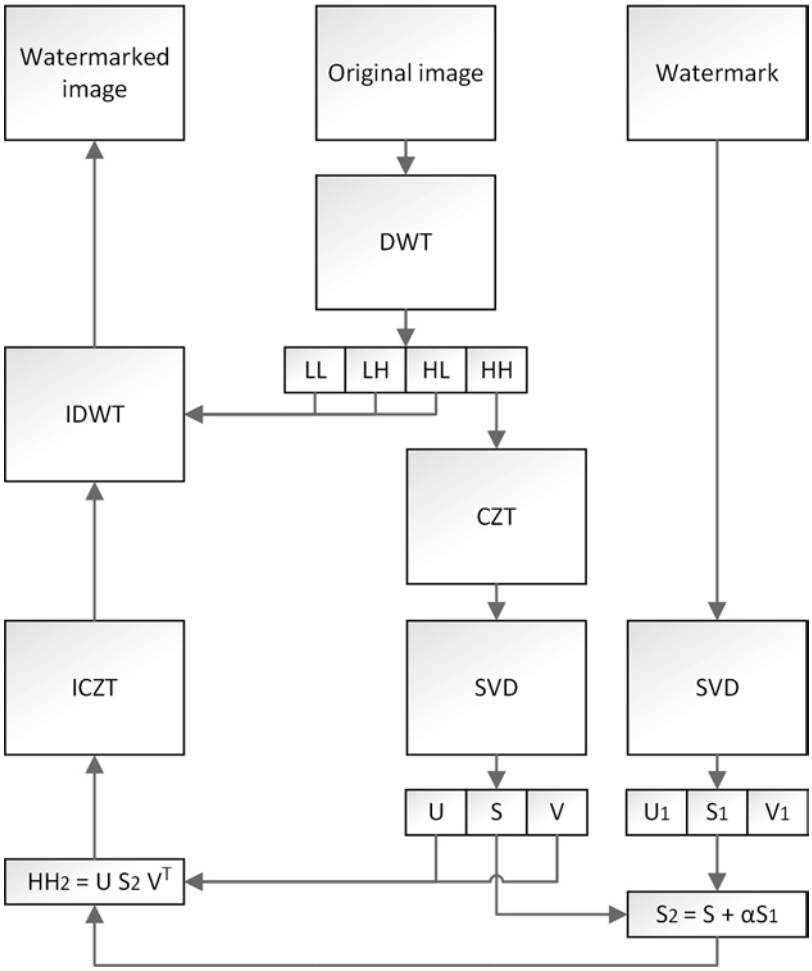


Figure 4. Mary Agoyi et al. embedding algorithm flowchart.

In order to extract watermark from image DWT is applied to original and watermarked image. Then CZT is used on both image's high-frequency band. After that SVD is applied to both CZT results and to watermark itself. To obtain singular value of extracted watermark image, singular value of decomposed original image is subtracted from singular value of decomposed watermark image and the result is divided by scaling factor. The last step is to combine orthogonal matrices from watermark with calculated singular value to get extracted watermark image. Flowchart of watermark extraction is shown in Figure 5.

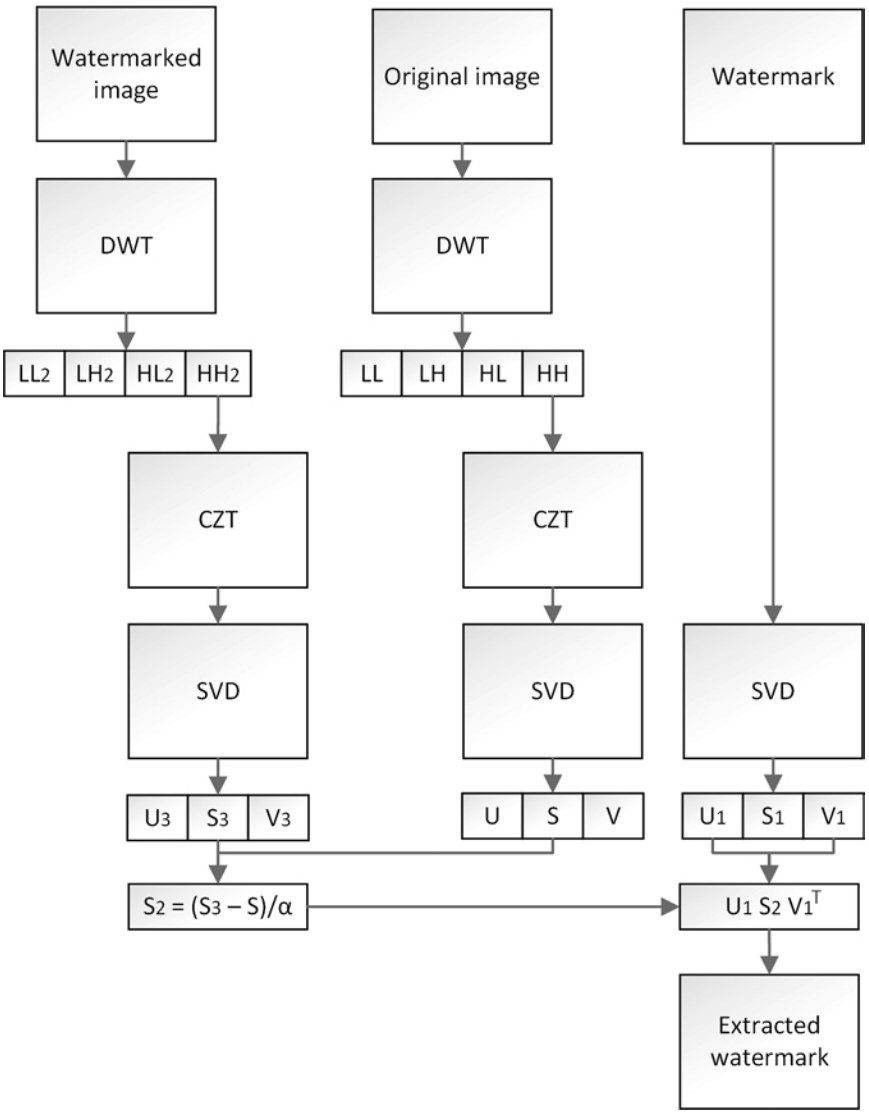


Figure 5. Mary Agoyi et al. extraction algorithm flowchart.

Proposed method showed better results in almost all cases, but there were some attacks where lay & Tsai method [36] performed better. In case of blurring and scaling attacks, proposed method had lower SSIM and correlation coefficient values. Also, when additive white Gaussian noise attack was used, almost all robustness measures had lower values that compared method. Taking into account that maximum value of SSIM and correlation coefficient is one and most of proposed methods values for these metrics were under 0.9, this shows that proposed method did not perform extremely well and there is room for great enhancements.

3.4. Adaptive Watermarking and Tree Structure Based Image Quality Estimation

Sha Wang *et al.* proposed a watermarking method [48] based on image quality estimation. In this scheme embedded watermark is used to estimate the degradation of host image under various distortions. DWT domain is used for watermark embedding. Set Partitioning in Hierarchical Trees (SPIHT) is used to categorize correlated DWT coefficients over DWT subbands and afterwards SPIHT trees are decomposed into a set of bit-planes. Watermark is added to the chosen bit-planes of selected DWT coefficients of chosen tree. Strength of the

watermark embedding is calculated by pre-analysing image content complexity in spatial domain and the perceptual masking effect of image gotten from DWT in frequency domain. Proposed method takes advantage of combining DWT and SPIHT what provides a novel summarization of local regional attributes of an image. Proposed method has many advantages. It is computationally efficient, quality loss of original image during watermark embedding is very small and it can assess image quality with good accuracy.

Watermark embedding algorithm starts with applying 3-level DWT to host image to obtain ten subbands. Watermark was embedded based on tree structure with adaptive embedding strength. Lastly, inverse 3-level DWT is used and watermark image is obtained. Tree structure based watermark embedder used in the second step is intended to insert the watermark binary bits into the given bit-planes of the chosen DWT coefficients of selected tree. This watermark embedder has three purposes: it has to form a tree structure, select the DWT coefficients and the trees for watermark embedding, and insert the watermark binary to the selected bit-planes of chosen coefficients. Described embedding scheme is shown in Figure 6. This figure shows watermark embedding together with watermark pre-processing and image pre-analysis.

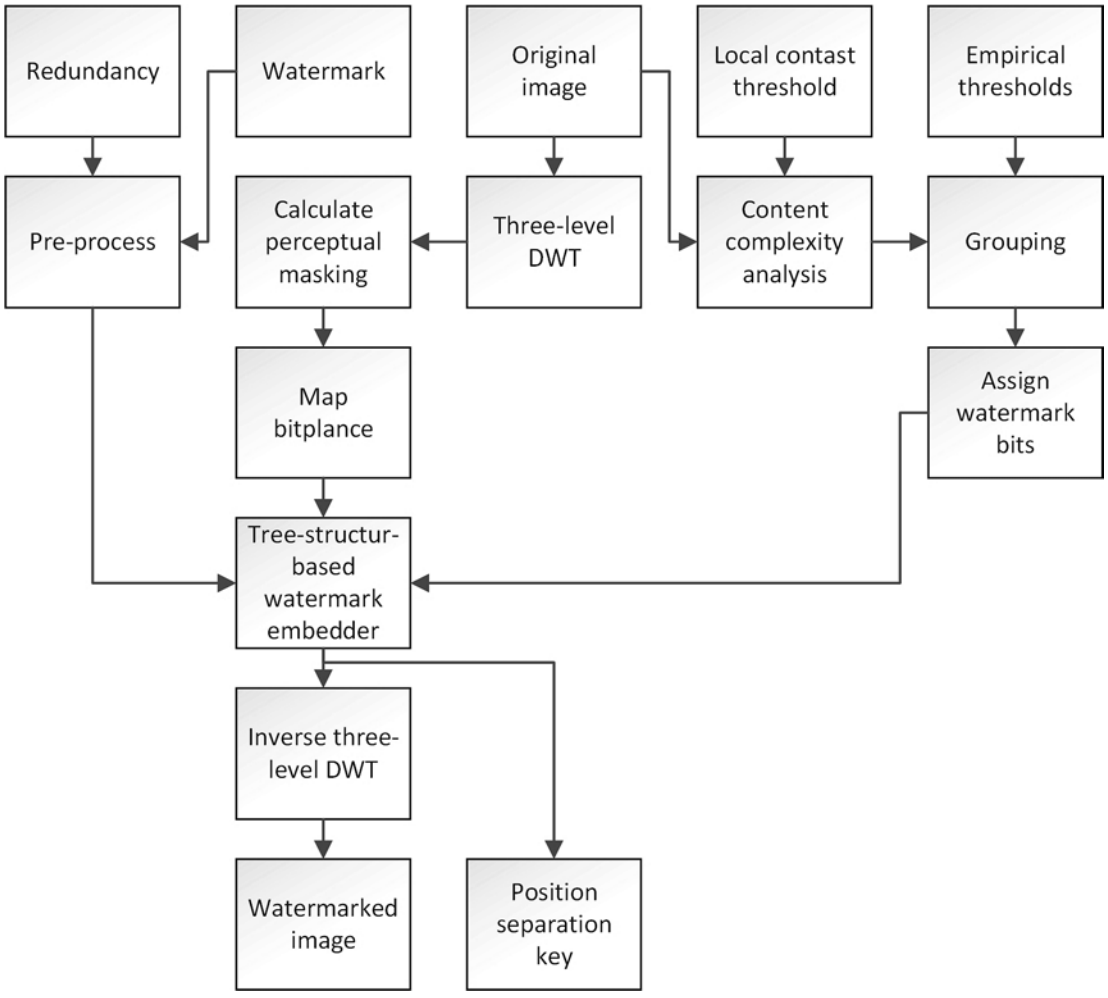


Figure 6. Sha Wang et al. method's watermark embedding process.

To extract, watermark sender send image group index what is used in watermark bit assignment. After calculating the human visual system masks of attacked watermark image the bit-plane indices for watermark extraction are obtained. Using subbands from

watermarked image 3-level DWT, position separation key, selected bit-planes and assigned bits, extracted watermark sequence is obtained. Figure 7 shows watermark extraction procedure.

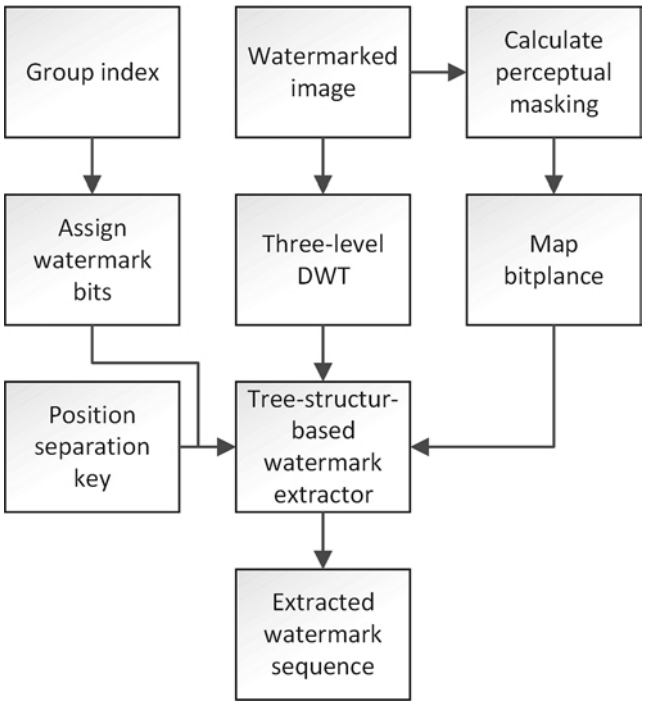


Figure 7. Sha Wang et al. method's watermark extraction process.

Authors conducted series of experiments. About 150 grayscale images with different textures like portraits, plants, animals and more were used as cover images. For distorting watermark images they used JPEG compression, JPEG2000 compression, Gaussian noise addition and Gaussian low-pass filtering with different distortion strengths. They did 16 sets of experiments where 100 images were in each set. From experimental results authors found out that proposed scheme has a very good imperceptibility result, average PSNR value of tested 150 images was 48.1776 dB. Also proposed method can evaluate image quality with good accuracy and it is computationally efficient.

3.5. A DCT-based digital watermarking algorithm for image

Jiang Yong Zheng *et al.* proposed digital watermarking algorithm [49] based on DCT. They claim that it is not only robust against signal processing attack, but it has larger watermarking capacity than other DCT-based watermarking algorithms. Their proposed method segments cover image into square blocks with size 8 pixels and each block is transferred to DCT domain. Watermark information is embedded and extracted using Global DCT Domain Watermarking Algorithm. Watermark is embedded into first L largest DCT coefficients, where L is positive number.

Embedding algorithm for proposed method starts with dividing host image into 8×8 blocks. After that DCT is applied to each block and L largest coefficients are found. Afterwards watermark image is multiplied with scaling factor α and multiplied with corresponding blocks of DCT L largest coefficients. Lastly inverse DCT is applied to all blocks and blocks are

added back together to get watermarked image. Flowchart of embedding algorithm is shown in Figure 8.

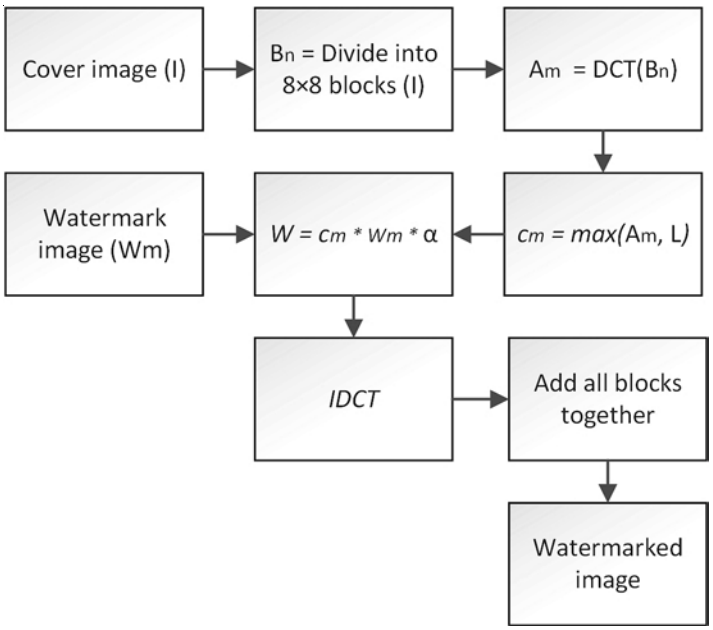


Figure 8. Jiang Yong Zheng et al. method's watermark embedding flowchart.

In order to extract watermark from watermarked image, cover image and watermarked image are divided into 8 × 8 blocks and DCT coefficients are found for all blocks. Following is done for all blocks. Find *L* largest DCT coefficients from original images block. Divide each *L* largest coefficient from original image divided by scaling factor α from corresponding watermarked images DCT coefficients. Extracted watermark images are gotten from the outcome of the previous step. Flowchart of watermark extraction is shown in Figure 9.

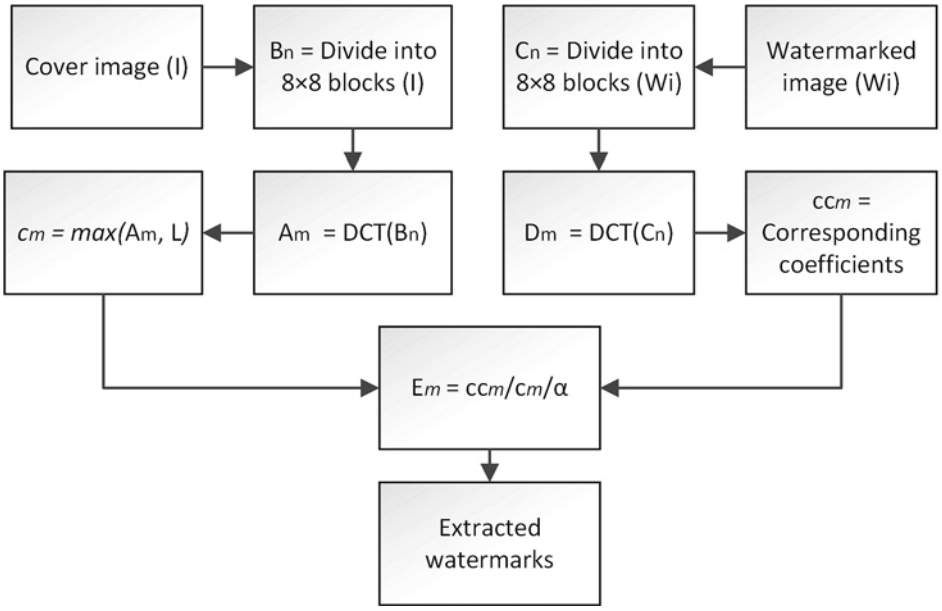


Figure 9. Jiang Yong Zheng et al. method's watermark extraction flowchart.

Authors of proposed method did analyses and found out that the information capacity what can be embedded into image is higher than in Block DCT Domain Algorithm and in Global

DCT Domain Algorithm. Proposed method's capacity is $N_1 * N_2 * L/64$ while Block DCT Domain Algorithm had $N_1 * N_2/64$ and Global DCT Domain Algorithm has only L, where $N_1$is the width of cover image, N_2 is height of cover image and L is the number of largest coefficients what were taken from DCT result.

4. Proposed watermarking schemes

4.1. Grayscale image watermarking

Many new watermarking techniques proposed nowadays use grayscale images. Grayscale images are so widely used because they are easier to implement and to maintain. Grayscale watermarking techniques can be easily expanded to colour images and even to videos and other similar signals, because the core functionality does not change when we add colour channels to image or when sequence of images are used. In this section I try to find the best watermarking scheme for grayscale images what could be later extended to colour images.

4.1.1. Watermarking technique using orthogonal-triangular decomposition

Proposed watermark scheme using orthogonal-triangular decomposition is divided into two steps, first is watermark embedding and the second is watermark extraction. This watermarking method will embed watermark into low entropy blocks of image, taking advantage of the best characteristics of discrete wavelet transform, chirp z-transform, singular value decomposition and orthogonal-triangular decomposition also known as QR decomposition. In the following subsections, you can see details of these steps.

4.1.1.1. Watermark embedding

Flowchart of embedding algorithm is presented in Figure 10, pseudo code of given embedding method is shown in Listing 1 and it is described below:

1. Divide the original *Sx* x *Sy* image into $\alpha*\beta$ blocks. Let $M=Sx/\alpha$ and $N=Sy/\beta$. Then each block can be represented as in Equation 5.

$$B_{mn} \quad m \in \{1 \dots M\}, n \in \{1 \dots N\}$$

Equation 5

2. Find the entropy value for each block. Where the entropy value is denoted by *E*.

3. Find the average entropy values of all blocks and set that value as the threshold *t*. *t* can be calculated as given in Equation 6.

$$t = \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{E(B_{mn})}{M * N}$$

Equation 6

4. DWT is applied to each block with entropy value less than the threshold to decompose it into subbands as given in Equation 7.

$$LL_{mn}LH_{mn}HL_{mn}HH_{mn} = DWT(B_{mn}), \forall B_{mn} \in \{B_{mn}: E(B_{mn}) < t\}$$

Equation 7

5. Compute the CZT of the low-frequency subband LL_{mn} for each decomposed block as given in Equation 8.

$$C_{mn} = CZT(LL_{mn})$$

Equation 8

6. Apply QR decomposition to matrix C_{mn} from Equation 8 to further decompose it as follows in Equation 9.

$$[Q_{mn}R_{mn}] = QR(C_{mn})$$

$$D_{1m} = \text{diag}(R_{mn})$$

$$D_{mn} = \text{zeros}(R_{mn})$$

$$D_{mm} = D_{1m}$$

Equation 9

7. SVD is used on diagonal matrix D_{mn} from Equation 9 to further decompose it as follows in Equation 10.

$$[U_{mn}S_{mn}V_{mn}] = SVD(D_{mn})$$

Equation 10

8. Apply SVD to watermark image W to decompose it as follows Equation 11.

$$[U_1S_1V_1] = SVD(W)$$

Equation 11

9. Update the singular value of the decomposed image with the singular value of the watermark image using a scaling factor γ to be inserted. γ controls the strength of the watermark. This is given in Equation 12.

$$S_{2mn} = S_{mn} + \gamma S_1$$

Equation 12

10. Combine the orthogonal matrixes of the decomposed original image from Equation 10 with the modified singular value matrix as given in Equation 13.

$$D_{1mn} = U_{mn} * S_{2mn} * V_{mn}^T$$

Equation 13

11. Replace upper-triangular matrix R_{mn} diagonal values with the modified D_{1mn} as given in Equation 14.

$$R_{mm} = D_{1mm}$$

Equation 14

12. Combine unitary matrix Q_{mn} with watermarked upper-triangular matrix R_{mn} as given in Equation 15

$$I_{mn} = Q_{mn} * R_{mn}$$

Equation 15

13. Compute the inverse CZT of I_{mn} to get the modified low-frequency subband as given in Equation 16.

$$LL_{2mn} = ICZT(I_{mn})$$

Equation 16

14. Apply the inverse DWT to the decomposed images, using the modified LL_{2mn} instead of LL_{mn} to get the watermarked image block, as shown in Equation 17.

$$I_{2mn} = IDWT(LL_{2mn}\ LH_{mn}\ HL_{mn}\ HH_{mn})$$

Equation 17

15. Combine watermarked low entropy image blocks with high entropy blocks to get watermarked image.

Note that steps 4 to 14 are applied to all blocks with entropy values lower than the threshold.

```
READ OriginalImage
READ Watermark

FOR i = 1 : 4
        Blocks(i) = Get image block(OriginalImage, i)
        Entropy(i) = Find entropy(Blocks(i))
ENDFOR

AverageEntropy = SUM(Entropy)/i

FOR i = 1 : 4
        IF Entropy(i) < AverageEntropy
                [LL,LH,HL,HH] = Discrete wavelet transform(Blocks(i))
                Czt = Chirp Z-transform(LL)
                [Q,R] = QR decomposition(Czt)
                D = Find diagonal matrix(R)
                [U,S,V] = Singular value decomposition(D)
                [U1,S1,V1] = Singular value decomposition(Watermark)
                S_watermarked = S + S1 * ScalingFactor
                D = Inverse Singular value decomposition(U, S_watermarked, V)
                QR = Inverse QR decomposition(Q, R, D)
                LL = Inverse Chirp Z-transform (Czt, QR)
                Blocks(i) = Inverse Discrete wavelet transform(LL, LH, HL, HH)
        ENDIF
ENDFOR
WatermarkedImage = Add blocks together(Blocks)
```

Listing 1. QR decomposition based grayscale watermark embedding pseudocode.

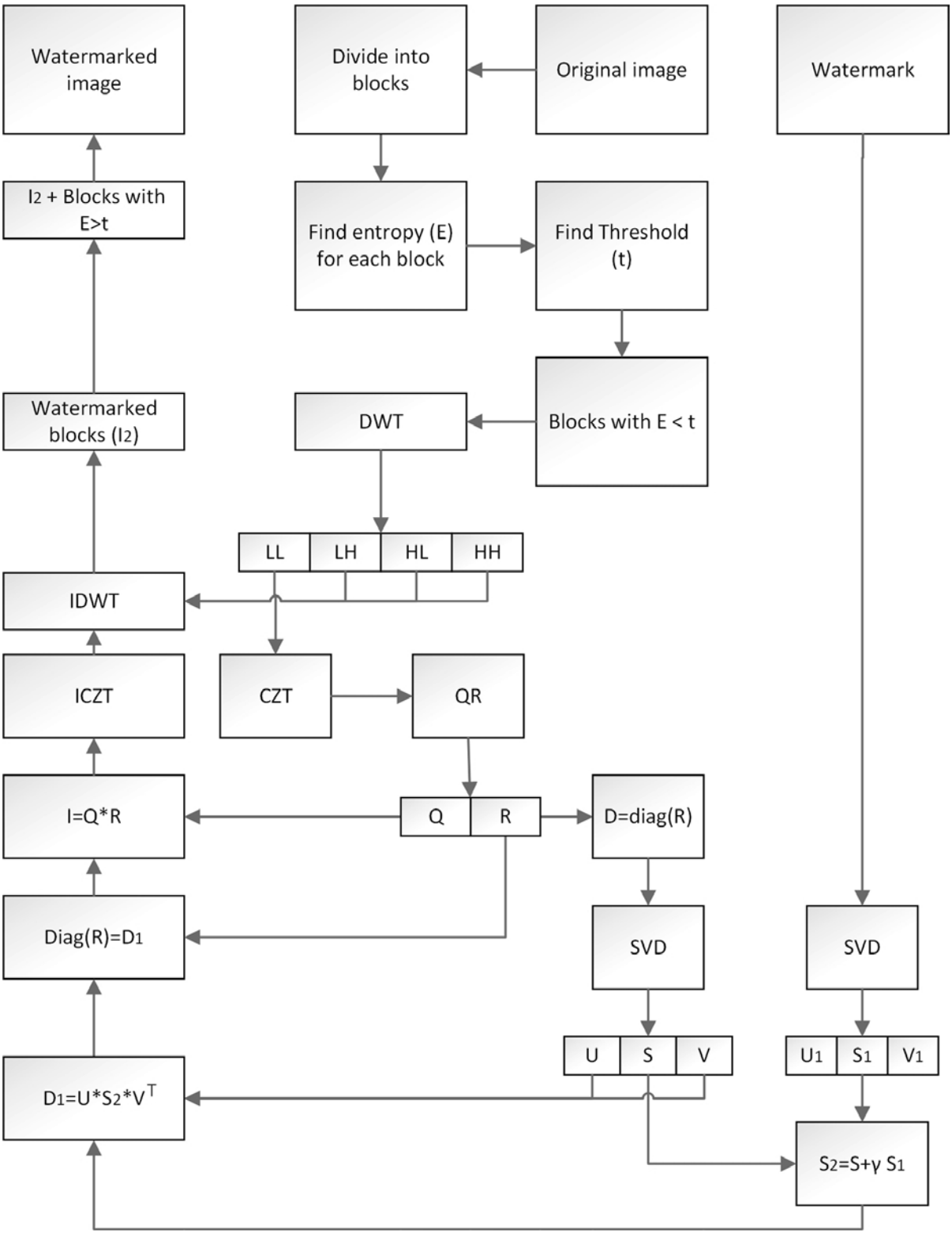


Figure 10. QR decomposition based grayscale watermark embedding algorithm.

4.1.1.2. Watermark extraction

Flowchart of extraction algorithm is presented in Figure 11. Watermark extraction algorithm is presented in Listing 2. Watermark extraction first stage has the same procedure that watermark embedding has, only it is performed on the watermarked image. For the original

image we also apply the methodology from Equation 5 to Equation 10. Equation 11 is also applied to watermark image. Singular values of original image blocks are then subtracted from singular values of watermarked image block and the outcome is divided by scaling factor γ in order to generate the singular values of extracted watermark image as shown in Equation 18.

$$BS_{mn} = (S'_{mn} - S_{mn})/\gamma$$

Equation 18

Afterwards orthogonal matrixes U_1 and V_1 of the watermark image are combined with the obtained BS_{mn} to get the extracted watermark image for each block, this is given in Equation 19.

$$Wb = U_1 BS_{mn} V_1^T$$

Equation 19

The image is changed into black and white image by using the average of the image as the threshold.

```
READ OriginalImage
READ WatermarkedImage
READ Watermark
FOR i = 1 : 4
        Blocks(i) = Get image block(OriginalImage, i)
        Entropy(i) = Find entropy(Blocks(i))
        W_Blocks(i) = Get image block(WatermarkedImage, i)
ENDFOR
AverageEntropy = SUM(Entropy)/i
FOR i = 1 : 4
        IF Entropy(i) < AverageEntropy
                [LL,LH,HL,HH] = Discrete wavelet transform(Blocks(i))
                [LL',LH',HL',HH'] = Discrete wavelet transform(W_Blocks(i))
                W_Czt = Chirp Z-transform(LL')
                [Q,R] = QR decomposition(LL)
                D = Find diagonal matrix(R)
                [Q',R'] = QR decomposition(W_Czt)
                D' = Find diagonal matrix(R')
                [U,S,V] = Singular value decomposition(D)
                [U',S',V'] = Singular value decomposition(D')
                [U1,S1,V1] = Singular value decomposition(Watermark)
                S_Extracted = (S' - S) / ScalingFactor
                ExtractedWatermark = U1 * S_Extracted * Transpose(V1)
                ExtractedWatermark = Convert into black-and-
                        white(ExtractedWatermark)
        ENDIF
ENDFOR
```

Listing 2. QR decomposition based grayscale watermark extraction pseudocode.

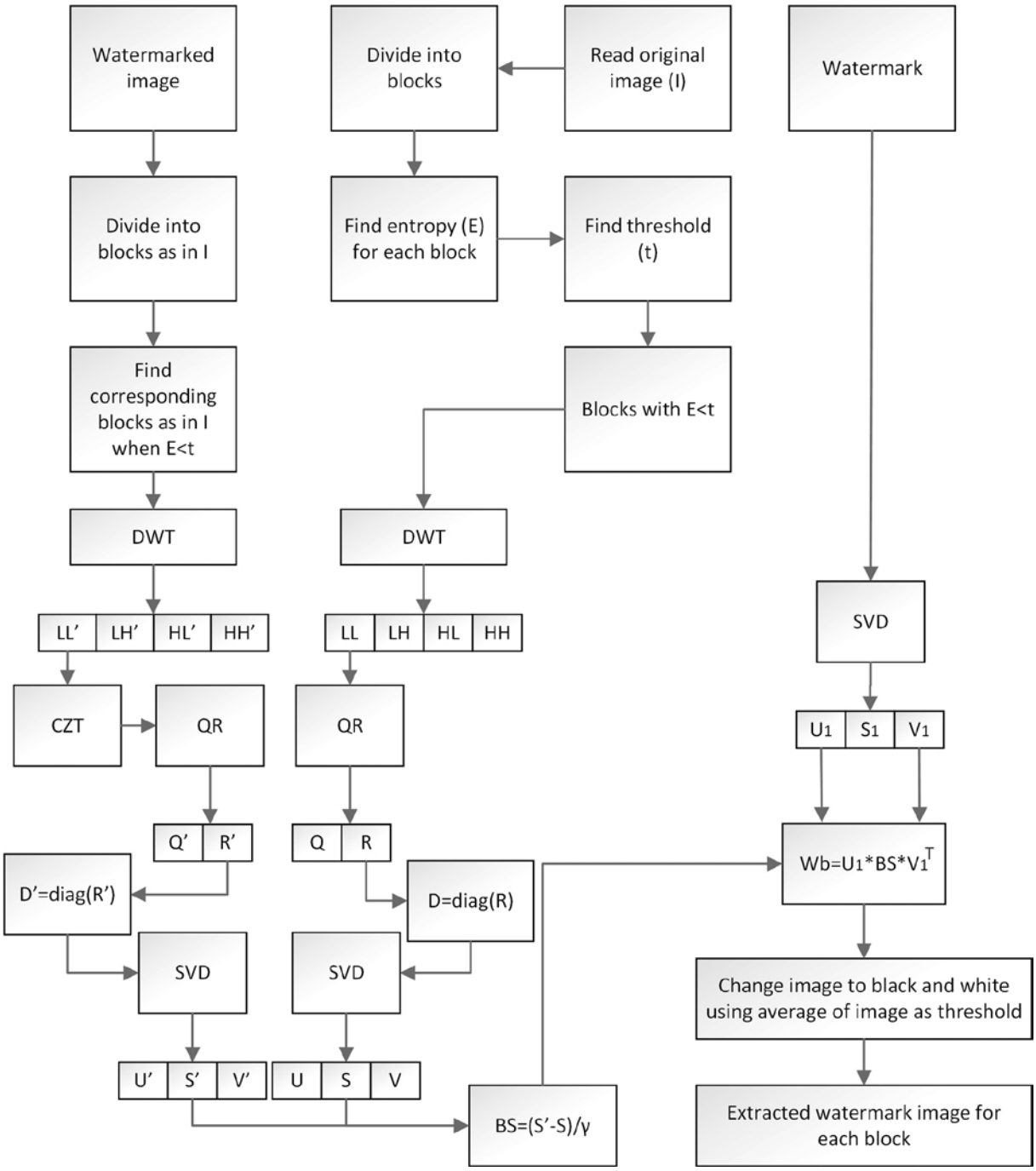


Figure 11. QR decomposition based grayscale watermark extraction algorithm.

4.1.2. Watermarking technique using lower and upper decomposition

Watermarking scheme using lower and upper decomposition is divided into embedding and extraction parts. The proposed watermarking algorithm embeds the watermark image into low entropy blocks of the host image. It utilizes discrete wavelet transform, chirp z-transform, singular value decomposition, and lower and upper decomposition and their characteristics will be described in this section. Detailed descriptions about both parts are presented in following subsections.

4.1.2.1. Watermark embedding

A step by step procedure of the embedding algorithm is illustrated in Figure 12, pseudocode of given embedding method is shown in Listing 3 and is explained below:

1. Divide the original *Sx* x *Sy* image into $\alpha*\beta$ blocks. Let $M=Sx/\alpha$ and $N=Sy/\beta$. Then each block can be represented as in Equation 5.
2. Find the entropy value for each block. Where the entropy value is denoted by *E*.
3. Find the average entropy values of all blocks and set that value as the threshold *t*. *t* can be calculated as given in Equation 6.
4. DWT is applied to each block with entropy value less than the threshold to decompose it into subbands as given in Equation 7.
5. Compute the CZT of the low-frequency subband LL_{mn} for each decomposed block as given in Equation 8.
6. Apply LUD to matrix C_{mn} from Equation 8 to further decompose it as follows in Equation 20.

$$[L_{mn}U_{2mn}] = LU(C_{mn})$$

$$U_{2mn} = D_{mn}U_{3mn}$$

$$C_{mn} = L_{mn}\, D_{mn}U_{3mn}$$

Equation 20

7. SVD is used on diagonal matrix D_{mn} from Equation 20 to further decompose it as follows in Equation 10.
8. Apply SVD to watermark image *W* to decompose it as follows Equation 11.
9. Update the singular value of the decomposed image with the singular value of the watermark image using a scaling factor γ to be inserted. γ controls the strength of the watermark. This is given in Equation 12.
10. Combine the orthogonal matrixes of the decomposed original image with the modified singular value matrix as given in Equation 13.
11. Combine the matrixes L_{mn} and U_{3mn} with the modified D_{1mn} matrix as given in Equation 21.

$$I_{mn} = L_{mn} * D_{1mn} * U_{3mn}$$

Equation 21

12. Compute the inverse CZT of I_{mn} to get the modified low-frequency subband as given in Equation 16.
13. Apply the inverse DWT to the decomposed images, using the modified LL_{2mn} instead of LL_{mn} to get the watermarked image block, as shown in Equation 17.

14. Combine watermarked low entropy image blocks with high entropy blocks to get watermarked image.

Note that steps 4 to 13 are applied to all blocks with entropy values lower than the threshold.

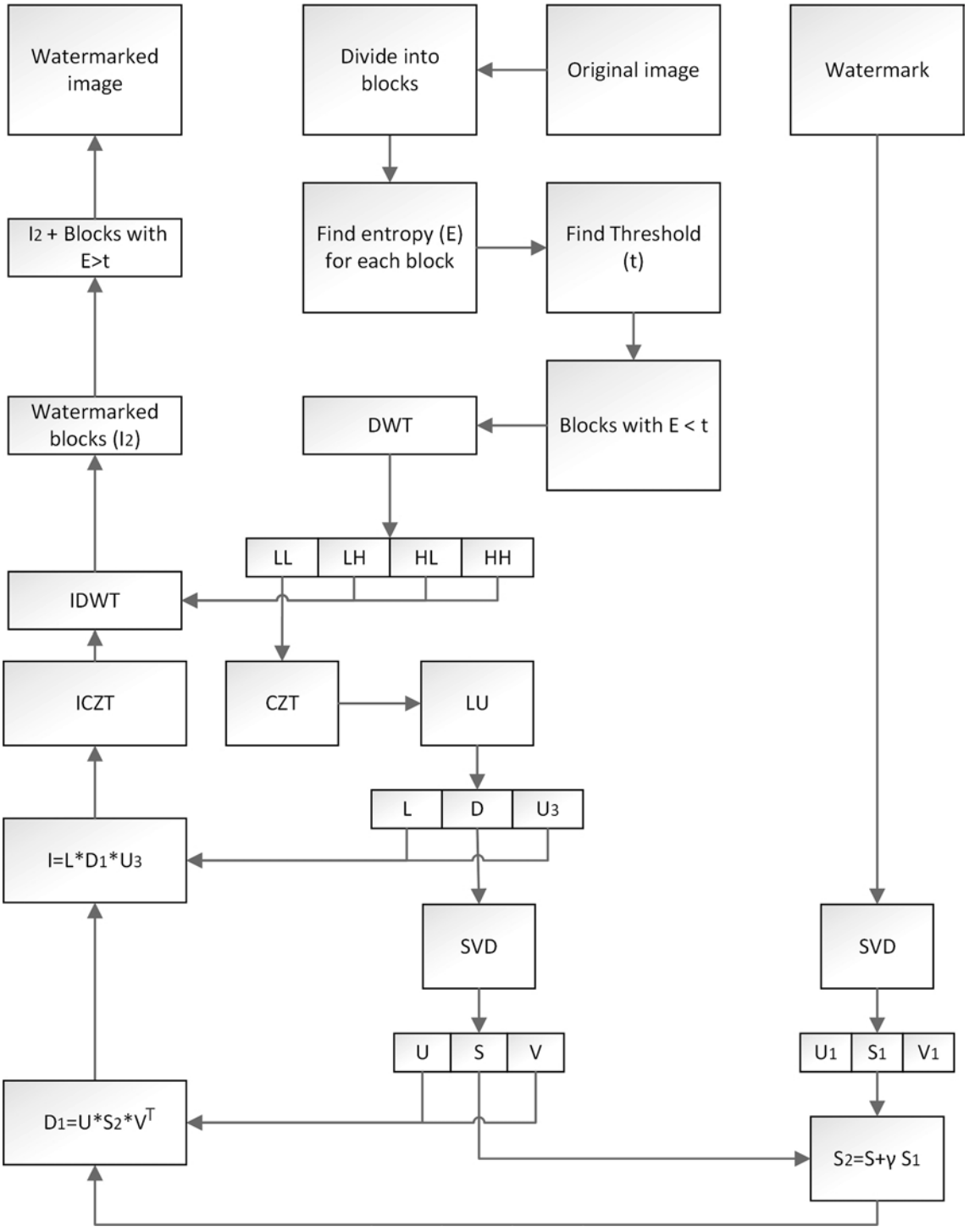


Figure 12. LU decomposition based grayscale watermark embedding algorithm.

```
READ OriginalImage
READ Watermark

FOR i = 1 : 4
        Blocks(i) = Get image block(OriginalImage, i)
        Entropy(i) = Find entropy(Blocks(i))
ENDFOR

AverageEntropy = SUM(Entropy)/i

FOR i = 1 : 4
        IF Entropy(i) < AverageEntropy
                [LL,LH,HL,HH] = Discrete wavelet transform(Blocks(i))
                Czt = Chirp Z-transform(LL)
                [L,U3] = LU decomposition(Czt)
                D = Find diagonal matrix(U3)
                [U,S,V] = Singular value decomposition(D)
                [U1,S1,V1] = Singular value decomposition(Watermark)
                S_watermarked = S + S1 * ScalingFactor
                D = Inverse Singular value decomposition(U, S_watermarked, V)
                LU = Inverse LU decomposition(L, U3, D)
                LL = Inverse Chirp Z-transform (Czt, LU)
                Blocks(i) = Inverse Discrete wavelet transform(LL,LH,HL,HH)
        ENDIF
ENDFOR
WatermarkedImage = Add blocks together(Blocks)
```

Listing 3. LU decomposition based grayscale watermark embedding pseudocode.

4.1.2.2. Watermark extraction

A step by step procedure of the extraction algorithm is illustrated in Figure 13, pseudocode of given extraction method is shown in Listing 4. Watermark extraction first stage is done similarly as is done in watermark embedding, only it is applied to watermarked image. Steps 1 to 7 from embedding algorithm are also applied to original image. Step 8 from embedding algorithm is applied for watermark image. Singular value of the decomposed blocks of the original image are subtracted from the singular value of the decomposed blocks of the watermarked image and the values are divided by the scaling factor γ to obtain the singular value of the watermark image. This is given in Equation 18. Lastly the orthogonal matrixes U_1 and V_1 of the watermark image are combined with the obtained BS_{mn} to get the extracted watermark image for each block, this is given in Equation 19. The image is changed into black and white image by using the average of the image as the threshold.

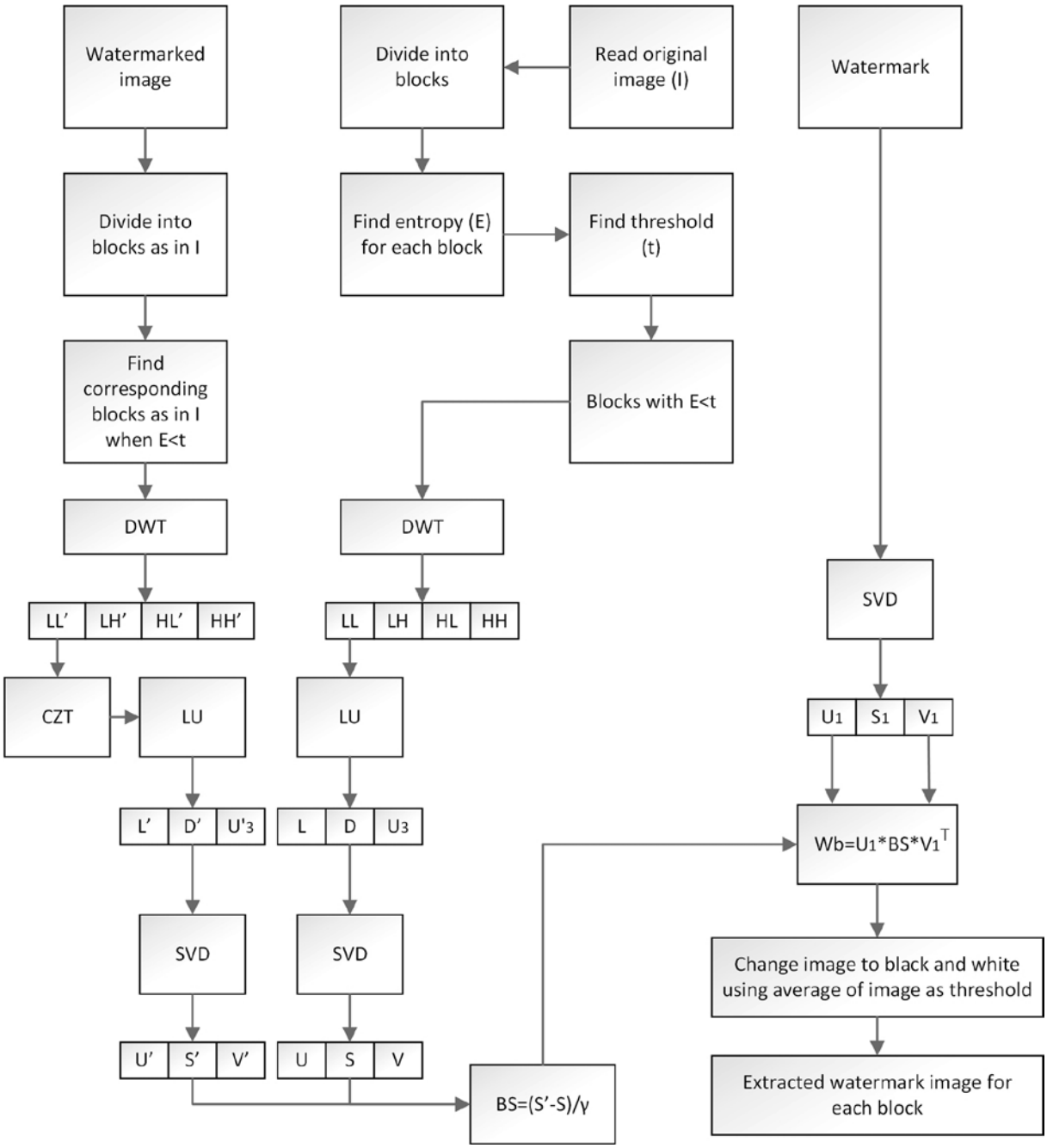


Figure 13. LU decomposition based grayscale watermark extraction algorithm.

```
READ OriginalImage
READ WatermarkedImage
READ Watermark

FOR i = 1 : 4
        Blocks(i) = Get image block(OriginalImage, i)
        Entropy(i) = Find entropy(Blocks(i))
        W_Blocks(i) = Get image block(WatermarkedImage, i)
ENDFOR

AverageEntropy = SUM(Entropy)/i

FOR i = 1 : 4
        IF Entropy(i) < AverageEntropy
                [LL,LH,HL,HH] = Discrete wavelet transform(Blocks(i))
                [LL',LH',HL',HH'] = Discrete wavelet transform(W_Blocks(i))
                W_Czt = Chirp Z-transform(LL')

                [L,U3] = LU decomposition(LL)
                D = Find diagonal matrix(U3)
                [L',U3'] = LU decomposition(W_Czt)
                D' = Find diagonal matrix(U3')

                [U,S,V] = Singular value decomposition(D)
                [U',S',V'] = Singular value decomposition(D')
                [U1,S1,V1] = Singular value decomposition(Watermark)

                S_Extracted = (S' - S) / ScalingFactor

                ExtractedWatermark = U1 * S_Extracted * Transpose(V1)
                ExtractedWatermark = Convert into black-and-
                        white(ExtractedWatermark)
        ENDIF
ENDFOR
```

Listing 4. LU decomposition based grayscale watermark extraction pseudocode.

4.2. Colour image watermarking

The proposed algorithm embeds watermark into colour image's three colour channels, red, green and blue (RGB). It uses entropy to find image blocks with low complexity and embeds watermark by combining best characteristics of DWT, CZT, LU decomposition and SVD. More detailed description with formulas and flowcharts of proposed embedding and extracting algorithm will be provided in this section.

4.2.1. Watermark embedding

Watermark embedding scheme is presented in Figure 14, pseudocode of given embedding method is shown in Listing 5 and explained in the following.

1. Divide original coloured cover image into three colour channels *R*, *G* and *B*. Apply following steps to each channel separately.
2. Divide $Sx \times Sy$ colour channel into $\alpha \times \beta$ blocks. Let $M=m/\alpha$ and $N=n/\beta$. Then each block can be described as in Equation 5.
3. Calculate entropy value for each block, where the entropy value is designated as *E*.
4. Calculate the average of all entropy values *E* from all blocks and denote the outcome as the threshold *t*. *t* can be calculated as given in Equation 6.
5. Use one-level DWT on each block with entropy value *E* less than calculated threshold *t* to decompose it into four subbands as given in Equation 7.
6. Calculate CZT of low-frequency subband LL_{mn} for all decomposed blocks as given in Equation 8.
7. Apply LU decomposition to matrix C_{mn} from Equation 8 to calculate diagonal matrix as given in Equation 20.
8. Apply SVD to diagonal matrix D_{mn} from Equation 20 to further decompose it as shown in Equation 10.
9. Apply SVD to watermark image *W* and decompose it as shown in Equation 11.
10. Calculate new singular values by adding original image's decomposed singular values to watermark image's singular values multiplied by scaling factor γ. γ is for controlling the strength of the added watermark. This is shown in Equation 12.
11. Combine unitary matrices $U_{mn}$and V_{mn} from decomposed original image with new singular values calculated in Equation 12 as shown in Equation 13.
12. Combine the matrices L_{mn} and U_{3mn} with modified D_{1mn} as shown in Equation 21.
13. Calculate inverse CZT of I_{mn} to get watermarked low-frequency subband as shown in Equation 16.
14. Calculate inverse DWT to get watermarked image block. Instead of LL_{mn} use modified LL_{2mn} as shown in Equation 17.
15. Add together modified low entropy blocks with high entropy blocks and all three colour channels to get watermarked colour image.

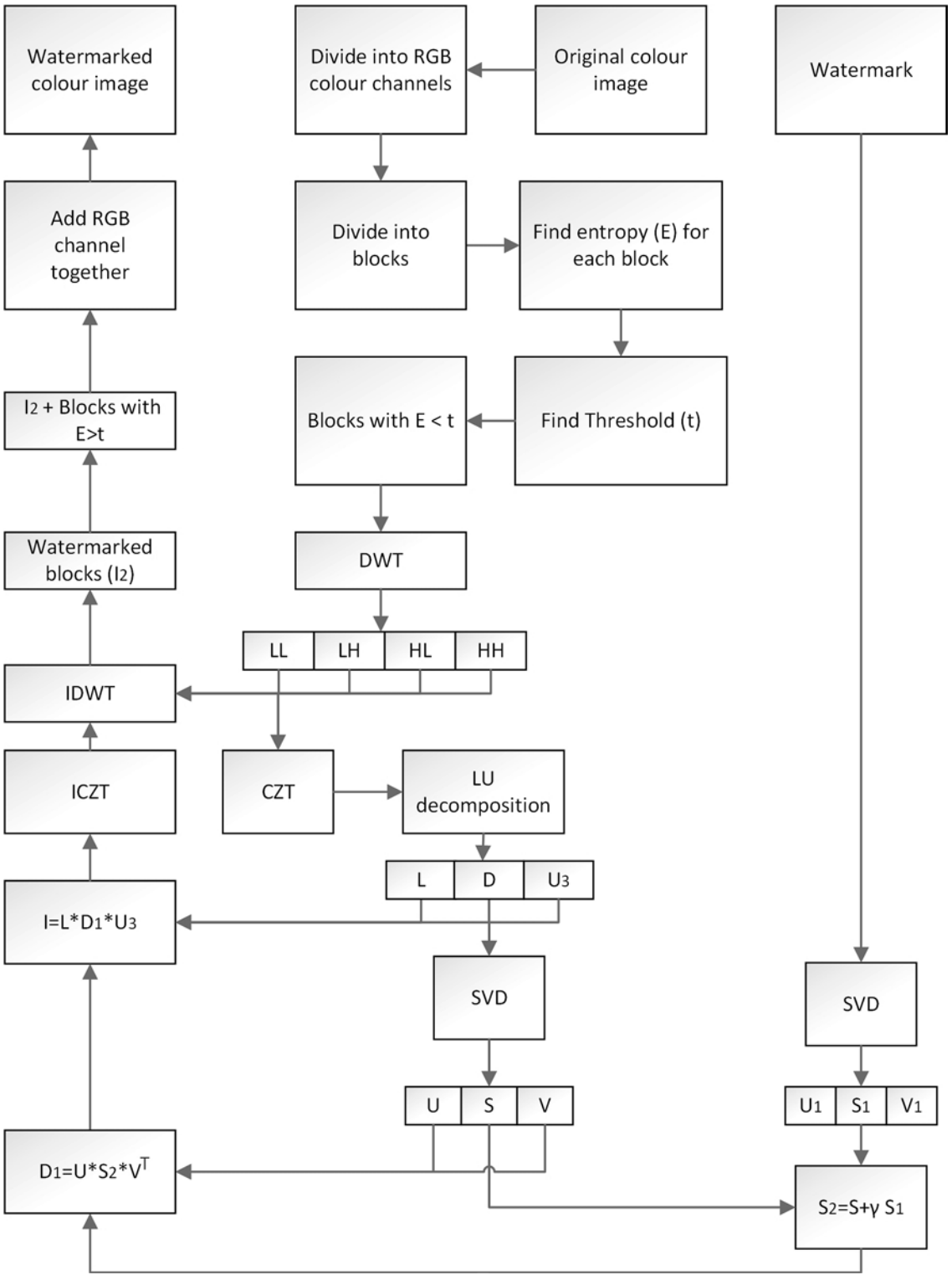


Figure 14. Colour image watermark embedding flowchart.

```
READ OriginalImage
READ Watermark

FOR j = 1 : 3
        Colour = Get colour matrix(OriginalImage, j)
        FOR i = 1 : 4
                Blocks(i) = Get image block(Colour, i)
                Entropy(i) = Find entropy(Blocks(i))
        ENDFOR

        AverageEntropy = SUM(Entropy)/i

        FOR i = 1 : 4
                IF Entropy(i) < AverageEntropy
                        [LL,LH,HL,HH] = Discrete wavelet transform(Blocks(i))
                        Czt = Chirp Z-transform(LL)
                        [L,U3] = LU decomposition(Czt)
                        D = Find diagonal matrix(U3)
                        [U,S,V] = Singular value decomposition(D)
                        [U1,S1,V1] = Singular value decomposition(Watermark)
                        S_watermarked = S + S1 * ScalingFactor
                        D = Inverse Singular value decomposition(U, S_watermarked,V)
                        LU = Inverse LU decomposition(L, U3, D)
                        LL = Inverse Chirp Z-transform (Czt, LU)
                        Blocks(i) = Inverse Discrete wavelet transform(LL, LH, HL, HH)
                ENDIF
        ENDFOR
        WatermarkedImages(j) = Add blocks together(Blocks)
ENDFOR
WatermarkedImage = Add colours together(WatermarkedImages)
```

Listing 5. Colour image watermark embedding pseudocode.

4.2.2. Watermark extraction

Watermark extraction scheme is presented in Figure 15, pseudocode of given extraction method is shown in Listing 6. Steps 1 to 8 from colour image embedding algorithm are also done in extraction algorithm for original image. In addition, the same procedure is applied to watermarked image. Step 9 is also applied to watermark image. Afterwards singular values of original image's block are subtracted from singular values of watermarked image's block and the outcome is divided by scaling factor γ to get singular values of extracted watermark image as shown in Equation 18. Unitary matrices $U_1$and V_1 from watermark image are combined with extracted singular values calculated in Equation 18 to get extracted watermark for each block as shown in Equation 19. Extracted watermark image is changed into black and white image by using average of the image as threshold.

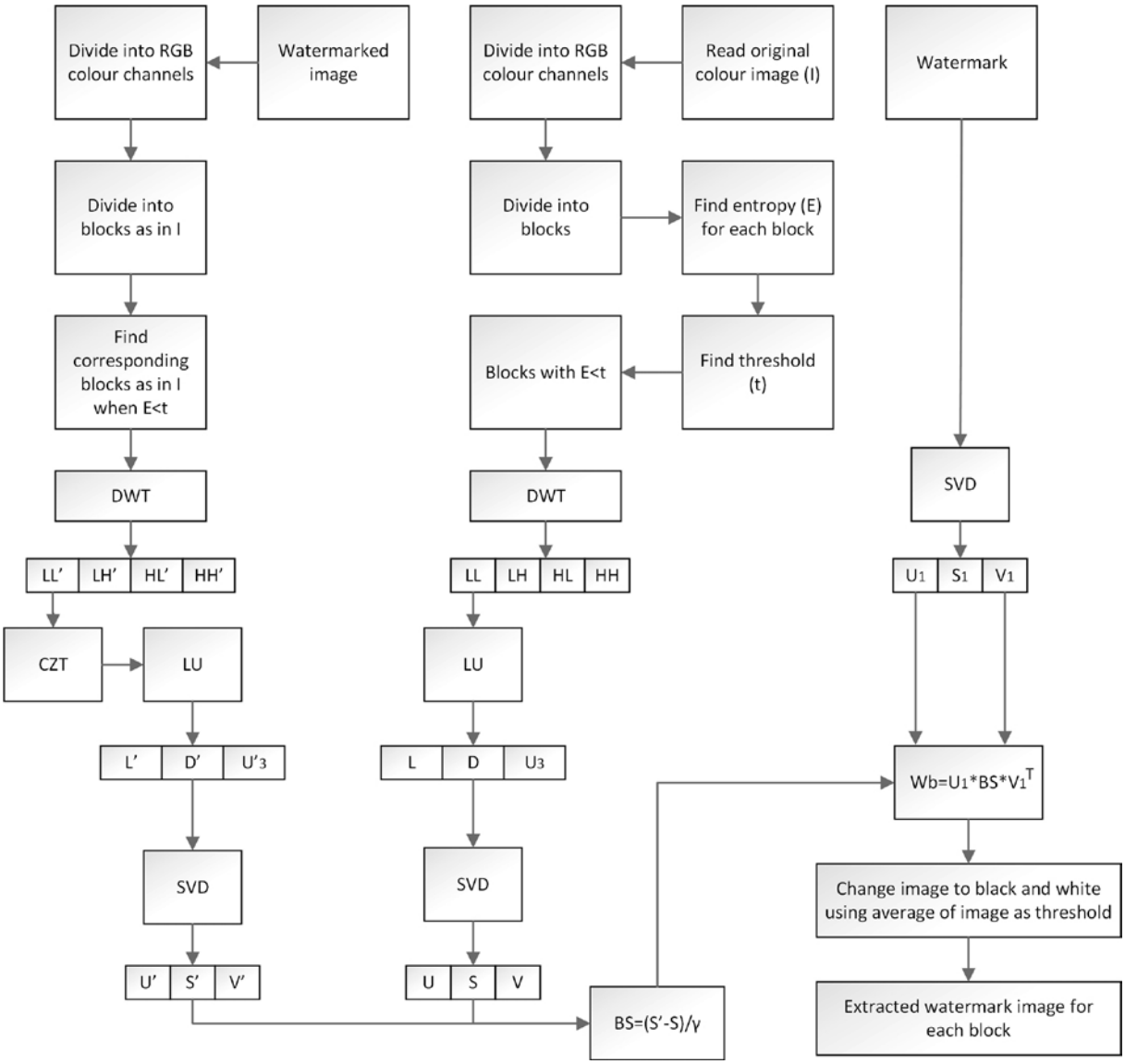


Figure 15. Colour image watermark extraction flowchart.

```
READ OriginalImage
READ WatermarkedImage
READ Watermark
FOR j = 1 : 3
        Colour = Get colour matrix(OriginalImage, j)
        FOR i = 1 : 4
                Blocks(i) = Get image block(Colour, i)
                Entropy(i) = Find entropy(Blocks(i))
                W_Blocks(i) = Get image block(WatermarkedImage, i)
        ENDFOR

        AverageEntropy = SUM(Entropy)/i

        FOR i = 1 : 4
                IF Entropy(i) <  AverageEntropy
                        [LL,LH,HL,HH] = Discrete wavelet transform(Blocks(i))
                        [LL',LH',HL',HH'] = Discrete wavelet transform(W_Blocks(i))

                        W_Czt = Chirp Z-transform(LL')

                        [L,U3] = LU decomposition(LL)
                        D = Find diagonal matrix(U3)
                        [L',U3'] = LU decomposition(W_Czt)
                        D' = Find diagonal matrix(U3')

                        [U,S,V] = Singular value decomposition(D)
                        [U',S',V'] = Singular value decomposition(D')
                        [U1,S1,V1] = Singular value decomposition(Watermark)

                        S_Extracted = (S' - S) / ScalingFactor
                        ExtractedWatermark = U1 * S_Extracted * Transpose(V1)
                        ExtractedWatermark = Convert into black-and-
                                white(ExtractedWatermark)
                ENDIF
        ENDFOR
ENDFOR
```

Listing 6. Colour image watermark extraction pseudocode.

5. Experimental results and discussion

5.1. Host and watermark images

In order to test proposed algorithms imperceptibility and robustness characteristics many colour and grayscale images were used. All used watermark images were grayscale images with resolution 128×128 pixels. Watermark images had different contexts like picture, writing with direction, logo and others. Figure 16 shows watermark images “Cameraman”, “Signature” and “UT Logo” used in experiments.

(a) Cameraman *(b) Signature* *(c) UT logo*

Figure 16. (a), (b) and (c) are three of the watermarks used.

For grayscale image watermarking, numerous well known benchmarks were used. All host images had 1024x1024 pixels resolution and they were pictures of people, nature, architecture, animals and others. Figure 17 contains cover images of “Lena”, “Barbara” and “Rose”, what were used in experiments presented in this works following sections.

(a) Lena *(b) Barbara* *(c) Rose*

Figure 17. (a), (b) and (c) are host images used in grayscale image watermarking algorithms.

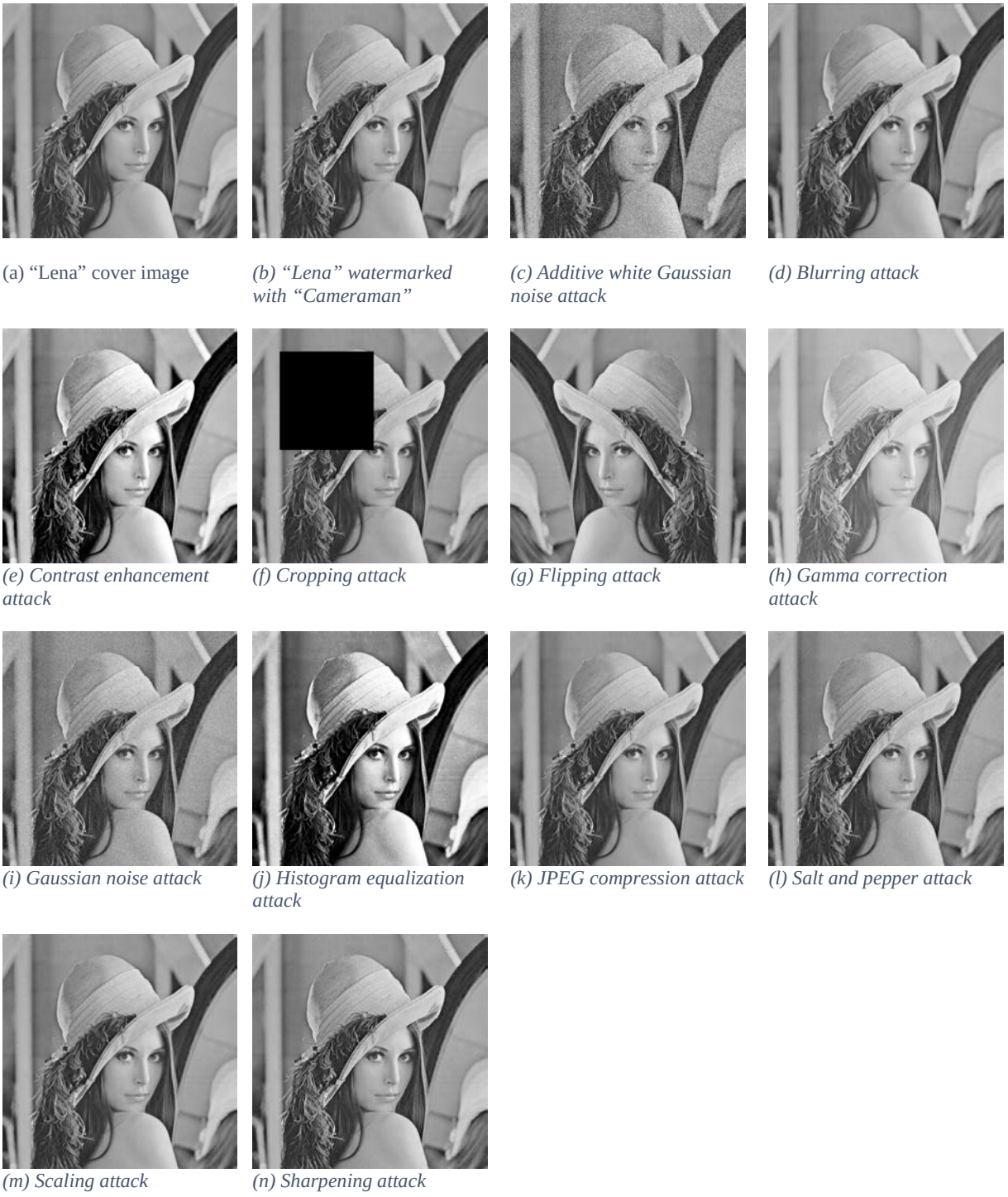
Cover images what were used in colour image watermarking experiments were well known benchmarks like “Lena”, “Barbara” and “Peppers. All host images were 1024x1024 pixel colour images. Figure 18 presents cover images “Lena”, “Barbara” and “Peppers” what were used in experiments presented in colour image watermarking section.

(a) Lena *(b) Barbara* *(c) Peppers*

Figure 18. (a), (b) and (c) are host images used in colour image watermarking algorithms.

5.2. Grayscale image watermarking

In this section two grayscale image watermarking technique's experimental results are presented. Proposed algorithms' imperceptibility and robustness are measured and compared with other state-of-the-art watermarking schemes. These quantitative results show that proposed method what uses lower and upper decomposition performs better than proposed method what uses orthogonal-triangular decomposition. Following subsections also show that proposed methods from this research outperforms other modern watermarking techniques.



(a) "Lena" cover image

(b) "Lena" watermarked with "Cameraman"

(c) Additive white Gaussian noise attack

(d) Blurring attack

(e) Contrast enhancement attack

(f) Cropping attack

(g) Flipping attack

(h) Gamma correction attack

(i) Gaussian noise attack

(j) Histogram equalization attack

(k) JPEG compression attack

(l) Salt and pepper attack

(m) Scaling attack

(n) Sharpening attack

Figure 19. (a) original image, (b) watermarked image, (c-n) watermarked image with different attacks applied to it.

Numerous tests were made with proposed methods and the results of this are presented in this paragraph. Figure 19 illustrates (a) “Lena” as host image, (b) host image with “Cameraman” as watermark and (c - n) the watermarked image after applying various attacks.

5.2.1. Watermarking technique using orthogonal-triangular decomposition

In order to compare proposed method’s imperceptibility qualities, PSNR metric was used. Table 1 shows comparison of PSNR values between Lai & Tsai method [45], Mary Agoyi *et al.* method [22] and proposed method. This table shows that proposed method maintains original image and there is no perceptual difference between watermarked and original image.

Table 1. Comparison of PSNR values between state-of-the-art methods and QR decomposition algorithm.

Host image	Watermark image	Lai & Tsai method [45]	Mary Agoyi *et al.* method [22]	QR decomposition method
Lena	Signature	26.3610	33.8945	85.1709
	Cameraman	28.1427	38.0844	86.7365
Barbara	Signature	25.0632	33.2083	107.5903
	Cameraman	26.7773	38.1450	107.5903
Rose	Signature	24.0930	30.0142	78.4658
	Cameraman	23.6368	34.1046	80.0337

In order to find out how well proposed algorithm satisfies robustness requirement, watermarked image was attacked with flipping, cropping, blurring, contrast enhancement, scaling, sharpening, Gaussian noise, additive white Gaussian noise, histogram equalization, Gamma correction, JPEG compression and salt & pepper noise attacks. Structure similarity ratio (SS ratio) and correlation coefficient values were measured to compare proposed method with other state-of-the-art methods. Experimental results show that proposed method is more robust than Lai & Tsai method [45] or Mary Agoyi *et al.* method [22] in the majority of attacks.

Table 2 shows comparison results when host image “Lena” is watermarked with “Signature”. This table shows that correlation coefficient and structure similarity ratio values are slightly lower for proposed method when flipping attack is used and they are significantly higher with all other attacks.

Table 3 presents correlation coefficient and structure similarity ratio results when different kinds of attacks are applied to host image “Lena” what is watermarked with “Cameraman”. Proposed method shows significantly higher correlation coefficient results when histogram equalization, contrast enhancement and scaling attacks are used. This table shows that proposed method has slightly better metric results when cropping, JPEG compression, salt and pepper noise, Gaussian noise, sharpening and additive white Gaussian noise attacks are applied to watermarked image. Structure similarity is slightly better when histogram equalization and scaling attacks are used. Correlation coefficient shows slightly better results when blurring attack is used. Structure similarity ratio is marginally worse for blurring attack for proposed method. For flipping and gamma correction attacks proposed method results are marginally worse.

Table 2. Comparison results for host image "Lena" watermarked with "Signature".

Attack	Lai & Tsai method [45]		Mary Agoyi *et al.* method [22]		QR decomposition method	
	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio
Flipping	1	1	1	1	0.6840	0.9293
Histogram equalization	0.3558	0.7162	0.5659	0.8986	0.7249	0.9430
Cropping	0.2406	0.5745	0.5464	0.8663	0.8535	0.9756
JPEG	0.2856	0.6329	0.5980	0.8943	0.8653	0.9780
Blurring	0.2885	0.6505	-0.4435	0.1659	0.4318	0.7770
Contrast enhancement	0.2902	0.6357	0.6963	0.9438	0.8550	0.9759
Salt and pepper noise	0.3243	0.6890	0.2723	0.7219	0.5189	0.8574
Gaussian noise	0.3123	0.6851	0.2639	0.7234	0.4439	0.7835
Sharpening	0.4717	0.8160	0.6306	0.9237	0.8104	0.9661
Gamma correction	0.3032	0.6423	0.4027	0.7564	0.7667	0.9551
Scaling	0.4148	0.7915	-0.4370	0.1718	0.7663	0.9550
AWGN	0.3423	0.7130	0.2767	0.7302	0.3992	0.7444

Table 3. Comparison results for host image "Lena" watermarked with "Cameraman".

Attack	Lai & Tsai method [45]		Mary Agoyi *et al.* method [22]		QR decomposition method	
	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio
Flipping	1	1	1	1	0.9924	0.9965
Histogram equalization	0.8869	0.9478	0.8713	0.9412	0.9945	0.9975
Cropping	0.8426	0.9257	0.8893	0.9491	0.9735	0.9876
JPEG	0.8968	0.9523	0.8837	0.9470	0.9625	0.9824
Blurring	0.6993	0.8569	-0.6327	0.1837	0.7000	0.8356
Contrast enhancement	0.9039	0.9556	0.8943	0.9518	0.9993	0.9997
Salt and pepper noise	0.8891	0.9512	0.6128	0.8089	0.9573	0.9844
Gaussian noise	0.8336	0.9219	0.5260	0.7510	0.9211	0.9609
Sharpening	0.9307	0.9677	0.8697	0.9404	0.9991	0.9996
Gamma correction	0.9302	0.9677	0.7609	0.8870	0.8586	0.9276
Scaling	0.7324	0.8752	-0.6313	0.1844	0.8791	0.9391
AWGN	0.7936	0.9008	0.5035	0.7413	0.8221	0.9143

Table 4 shows correlation coefficient and structure similarity metric results for Lai & Tsai method [45], Mary Agoyi *et al.* method [22] and for proposed schema when multiple attacks are applied on a host image "Rose" what is watermarked with "Cameraman".

Table 4. Comparison results for host image "Rose" watermarked with "Cameraman".

Attack	Lai & Tsai method [45]		Mary Agoyi *et al.* method [22]		QR decomposition method	
	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio
Flipping	1	1	1	1	1	1
Histogram equalization	0.9013	0.9547	0.7651	0.8893	0.9999	0.9999
Cropping	0.9336	0.9694	0.6467	0.8358	1	1
JPEG	0.9414	0.9733	0.8132	0.9143	1	1
Blurring	0.8354	0.9240	-0.6117	0.1948	0.9686	0.9853
Contrast enhancement	0.9337	0.9698	0.8572	0.9346	1	1
Salt and pepper noise	0.8532	0.9307	0.5886	0.7921	0.8866	0.9457
Gaussian noise	0.8206	0.9151	0.5306	0.7604	0.8274	0.9072
Sharpening	0.9738	0.9880	0.8496	0.9311	0.9713	0.9866
Gamma correction	0.9480	0.9763	0.8828	0.9462	1	1
Scaling	0.7767	0.8990	-0.6109	0.1952	1	1
AWGN	0.9212	0.9630	0.6668	0.8364	0.9058	0.9580

Table 5 presents comparison between proposed method and Lai & Tsai method [45] and Mary Agoyi *et al.* method [22]. In the table there are structure similarity and correlation coefficient metric results for many attacks.

Table 5. Comparison results for host image "Barbara" watermarked with "Cameraman".

Attack	Lai & Tsai method [45]		Mary Agoyi *et al.* method [22]		QR decomposition method	
	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio
Flipping	1	1	1	1	0.9992	0.9996
Histogram equalization	0.4989	0.7152	0.8795	0.9449	1	1
Cropping	0.9588	0.9812	0.5019	0.7606	1	1
JPEG	0.6115	0.7910	0.9367	0.9711	1	1
Blurring	0.8187	0.9178	-0.6214	0.1895	0.8624	0.9298
Contrast enhancement	0.5811	0.7701	0.8934	0.9514	1	1
Salt and pepper noise	0.6168	0.8038	0.6180	0.8107	0.9799	0.9892
Gaussian noise	0.6074	0.7870	0.5247	0.7558	0.8857	0.9452
Sharpening	0.8189	0.9122	0.8574	0.9348	0.9989	0.9995
Gamma correction	0.9122	0.9592	0.5670	0.7917	0.9995	0.9998
Scaling	0.6944	0.8560	-0.6196	0.1906	1	1
AWGN	0.7794	0.8952	0.5037	0.7486	0.8295	0.9108

Figure 20 shows extracted watermarks from host image “Lena” watermarked with “Cameraman”. This shows that extracted black and white watermarks are visually distinguishable for all tested attacks.

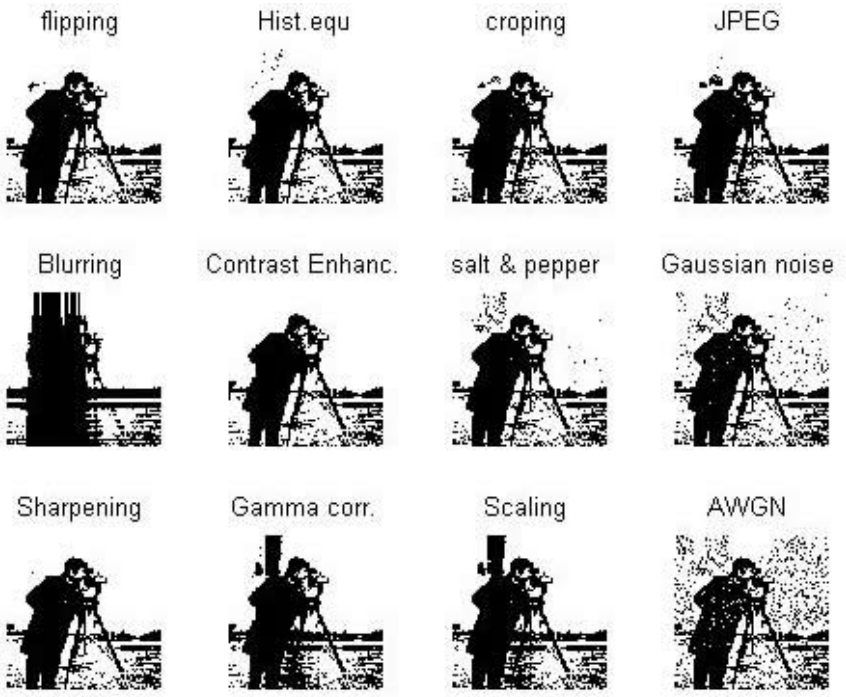


Figure 20. Extracted black and white watermarks from host image “Lena” watermarked with “Cameraman”.

5.2.2. Watermarking technique using lower and upper decomposition

In order to evaluate the visual quality of the images produced by the proposed algorithm what uses lower and upper decomposition, PSNR measure in dB is used. PSNR value of proposed method is compared with Lai & Tsai method [45] and with Mary Agoyi *et al.* proposed method [22] in Table 6.

Table 6. PSNR values of watermarked images for different watermarking methods.

Host image	Watermark image	Lai & Tsai method [45]	Mary Agoyi *et al.* method [22]	LU decomposition method
Lena	Signature	26.3610	33.8945	63.5570
	Cameraman	28.1427	38.0844	62.2561
Barbara	Signature	25.0632	33.2083	67.1622
	Cameraman	26.7773	38.1450	67.6618
Rose	Signature	24.0930	30.0142	57.3547
	Cameraman	23.6368	34.1046	59.9062

The robustness of the proposed method is tested by using various signal processing attacks, namely flipping, cropping, blurring, contrast enhancement, scaling, sharpening, Gaussian noise, additive white Gaussian noise, histogram equalization, Gamma correction, JPEG compression and salt & pepper noise on watermarked images. For comparison purposes with state-of-the-art algorithms, correlation coefficient and structure similarity ratio were used. In

Table 7, Lai & Tsai method [45] and Mary Agoyi *et al.* method [22] are compared with the proposed method using different host images, different watermarks and the previously referred attacks. These quantitative results show that proposed algorithm outperforms the aforementioned state-of-the-art algorithms. The proposed algorithm performs extremely well when cropping, JPEG compression, blurring, salt and pepper noise, Gaussian noise, scaling, gamma correction and adaptive white Gaussian noise attacks are used against the watermarked image.

Table 7 compares metric results between the proposed method and two state-of-the-art methods where "Lena" is the host image and "Signature" is the watermark. Table 7 shows that proposed method's correlation coefficient and structure similarity ratio are significantly higher when cropping, blurring, salt and pepper noise, Gaussian noise scaling, gamma correction, adaptive white Gaussian noise and JPEG compression attacks are performed. For histogram equalization and contrast enhancement attacks the proposed algorithm metrics are slightly better that other two methods results. When flipping and sharpening attacks are performed, the proposed method performs marginally worse than other two methods.

Table 7. Comparison results for host image "Lena" watermarked with "Signature".

Attack	Lai & Tsai method [45]		Mary Agoyi *et al.* method [22]		LU decomposition method	
	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio
Flipping	1	1	1	1	0.9059	0.9855
Histogram equalization	0.3558	0.7162	0.5659	0.8986	0.6140	0.9008
Cropping	0.2406	0.5745	0.5464	0.8663	0.9780	0.9969
JPEG	0.2856	0.6329	0.5980	0.8943	0.8883	0.9824
Blurring	0.2885	0.6505	-0.4435	0.1659	0.4329	0.7781
Contrast enhancement	0.2902	0.6357	0.6963	0.9438	0.7568	0.9524
Salt and pepper noise	0.3243	0.6890	0.2723	0.7219	0.5433	0.8537
Gaussian noise	0.3123	0.6851	0.2639	0.7234	0.4736	0.8148
Sharpening	0.4717	0.8160	0.6306	0.9237	0.5747	0.8811
Gamma correction	0.3032	0.6423	0.4027	0.7564	0.5511	0.8678
Scaling	0.4148	0.7915	-0.4370	0.1718	0.8913	0.9829
AWGN	0.3423	0.7130	0.2767	0.7302	0.4477	0.7737

Table 8 shows correlation coefficient and structure similarity ratio metrics results when several attacks are performed on the watermarked image where "Lena" is the host image and "Cameraman" is the watermark. The proposed method performs significantly better than the compared methods when salt and pepper noise, Gaussian noise, scaling and adaptive white Gaussian noise attacks are applied to given image. In the case of histogram equalization, cropping, JPEG compression, contrast enhancement and sharpening attacks, proposed method performs marginally better than other two algorithms. Mary Agoyi *et al.* method [22] shows

significantly worse result when blurring attack is used, but Lai & Tsai method [45] is slightly better than the proposed method. For gamma correction attack the proposed algorithm performs slightly better than Mary Agoyi *et al.* method [22] but results are marginally lower than Lai & Tsai method [45] results. For a flipping attack Table 8 shows that for given watermarked image the proposed method performs slightly worse than the other two methods.

Table 8. Comparison results for host image "Lena" watermarked with "Cameraman".

Attack	Lai & Tsai method [45]		Mary Agoyi *et al.* method [22]		LU decomposition method	
	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio
Flipping	1	1	1	1	0.9999	0.9999
Histogram equalization	0.8869	0.9478	0.8713	0.9412	0.9983	0.9992
Cropping	0.8426	0.9257	0.8893	0.9491	0.9367	0.9695
JPEG	0.8968	0.9523	0.8837	0.9470	0.9579	0.9801
Blurring	0.6993	0.8569	-0.6327	0.1837	0.6819	0.8277
Contrast enhancement	0.9039	0.9556	0.8943	0.9518	1	1
Salt and pepper noise	0.8891	0.9512	0.6128	0.8089	0.9852	0.9968
Gaussian noise	0.8336	0.9219	0.5260	0.7510	0.9604	0.9790
Sharpening	0.9307	0.9677	0.8697	0.9404	0.9999	0.9999
Gamma correction	0.9302	0.9677	0.7609	0.8870	0.8774	0.9382
Scaling	0.7324	0.8752	-0.6313	0.1844	0.8561	0.9262
AWGN	0.7936	0.9008	0.5035	0.7413	0.8888	0.9342

Table 9 shows the correlation coefficient and structure similarity ratio metrics when several attacks are applied to the watermarked image where "Rose" is used as the host image and "Cameraman" is the watermark.

Table 10 shows the correlation coefficient and structure similarity ratio between the original watermark and the extracted watermark after many signal processing attacks are used on the watermarked image. The watermarked image is obtained using "Barbara" as the host image and "Cameraman" as the watermark. This table shows that the proposed method outperforms the other two algorithms in almost all attacks.

In order to show the visual quality of proposed method, black and white pictures of the extracted attacked watermark "Cameraman" from the host image "Lena" are shown in Figure 21. The visual results show the retrieved watermarks are easily distinguishable, and the proposed algorithm preserves the watermark image after various signal processing attacks.

Table 9. Comparison results for host image "Rose" watermarked with "Cameraman".

Attack	Lai & Tsai method [45]		Mary Agoyi *et al.* method [22]		LU decomposition method	
	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio
Flipping	1	1	1	1	1	1
Histogram equalization	0.9013	0.9547	0.7651	0.8893	0.9967	0.9985
Cropping	0.9336	0.9694	0.6467	0.8358	1	1
JPEG	0.9414	0.9733	0.8132	0.9143	0.9997	0.9999
Blurring	0.8354	0.9240	-0.6117	0.1948	0.8537	0.9249
Contrast enhancement	0.9337	0.9698	0.8572	0.9346	0.9992	0.9996
Salt and pepper noise	0.8532	0.9307	0.5886	0.7921	0.8442	0.9570
Gaussian noise	0.8206	0.9151	0.5306	0.7604	0.7657	0.8885
Sharpening	0.9738	0.9880	0.8496	0.9311	0.9740	0.9879
Gamma correction	0.9480	0.9763	0.8828	0.9462	0.9988	0.9995
Scaling	0.7767	0.8990	-0.6109	0.1952	1	1
AWGN	0.9212	0.9630	0.6668	0.8364	0.8405	0.9501

Table 10. Comparison results for host image "Barbara" watermarked with "Cameraman".

Attack	Lai & Tsai method [45]		Mary Agoyi *et al.* method [22]		LU decomposition method	
	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio	Correlation coefficient	Structure similarity ratio
Flipping	1	1	1	1	0.9992	0.9996
Histogram equalization	0.4989	0.7152	0.8795	0.9449	1	1
Cropping	0.9588	0.9812	0.5019	0.7606	1	1
JPEG	0.6115	0.7910	0.9367	0.9711	1	1
Blurring	0.8187	0.9178	-0.6214	0.1895	0.8078	0.8976
Contrast enhancement	0.5811	0.7701	0.8934	0.9514	0.9999	0.9999
Salt and pepper noise	0.6168	0.8038	0.6180	0.8107	0.9855	0.9982
Gaussian noise	0.6074	0.7870	0.5247	0.7558	0.9547	0.9780
Sharpening	0.8189	0.9122	0.8574	0.9348	1	1
Gamma correction	0.9122	0.9592	0.5670	0.7917	1	1
Scaling	0.6944	0.8560	-0.6196	0.1906	1	1
AWGN	0.7794	0.8952	0.5037	0.7486	0.9015	0.9432

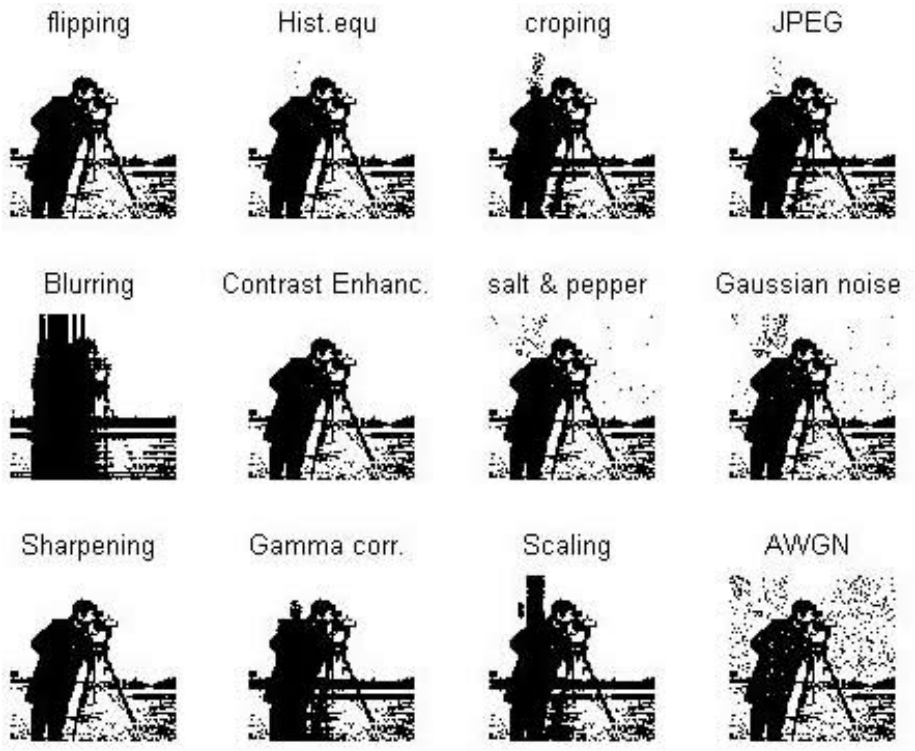


Figure 21. Black and white extracted watermarks.

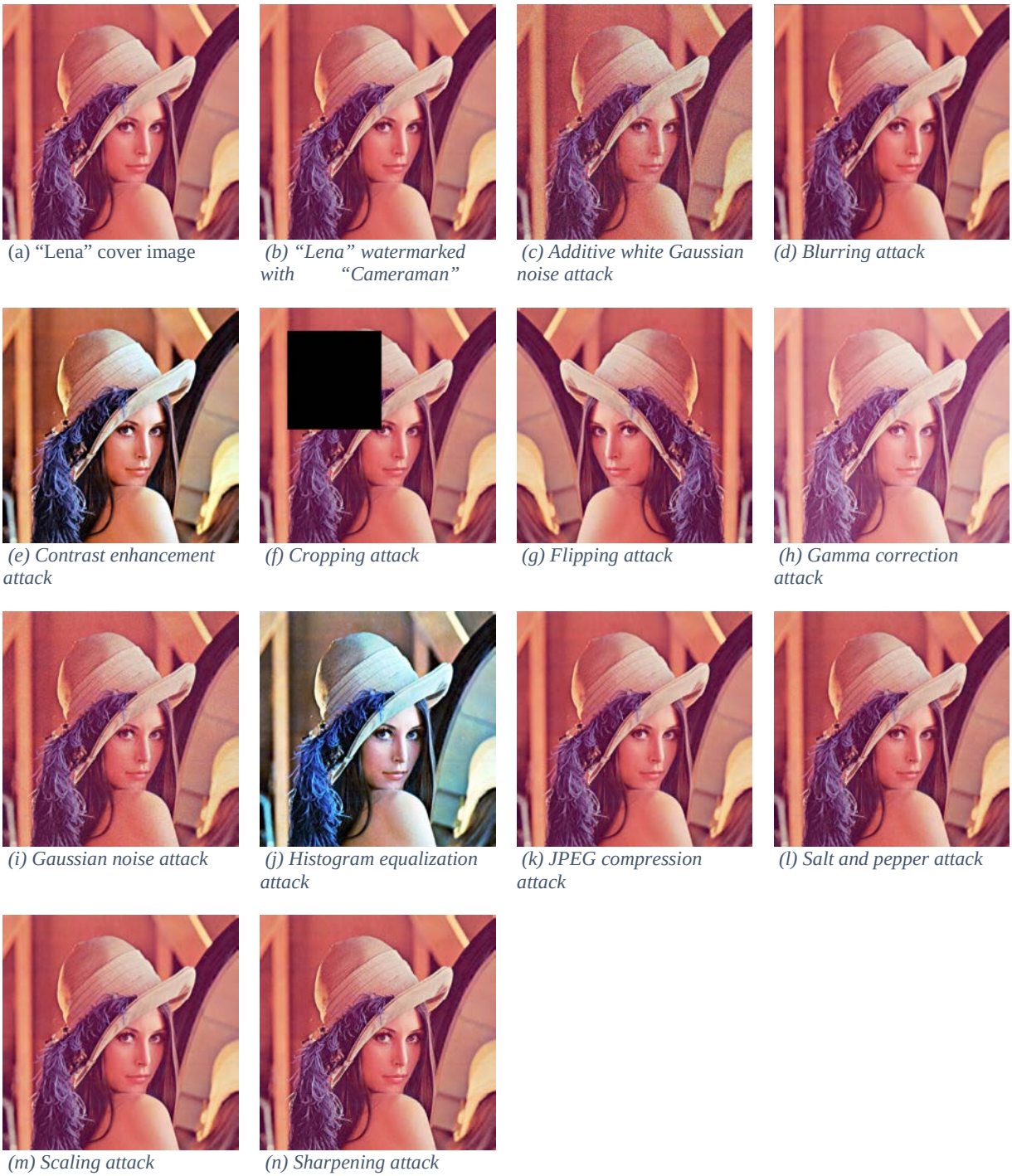
5.3. Colour image watermarking

Various experiments were conducted during this research, numerous host images were watermarked with different watermark images. Figure 22 presents (a) the host image “Lena”, (b) watermarked host image and (c - n) watermarked image with different attacks applied to it.

In order to evaluate imperceptibility characteristics, quality measurement PSNR was used. It measures image quality in decibels. PSNR values of proposed colour image watermarking method is compared with LSB method, Lai & Tsai proposed method [45] and Agoyi *et al.* proposed method [22], results of this comparison are shown Table 11.

Table 11. PSNR values of watermarked images for different watermarking methods.

Host image	Watermark image	LSB	Lai & Tsai method [45]	Mary Agoyi *et al.* method [22]	Proposed RGB method
Lena	Signature	50.87	26.36	33.89	60.97
	Cameraman	50.88	28.14	38.08	59.37
	UT logo	50.65	33.02	47.37	64.53
Barbara	Signature	50.80	25.06	33.21	60.73
	Cameraman	50.79	26.78	38.15	60.84
	UT logo	50.79	32.52	47.38	66.06
Peppers	Signature	50.12	25.12	35.89	62.42
	Cameraman	50.12	27.07	39.02	62.72
	UT logo	50.14	32.22	47.06	67.45



(a) “Lena” cover image

(b) “Lena” watermarked with “Cameraman”

(c) Additive white Gaussian noise attack

(d) Blurring attack

(e) Contrast enhancement attack

(f) Cropping attack

(g) Flipping attack

(h) Gamma correction attack

(i) Gaussian noise attack

(j) Histogram equalization attack

(k) JPEG compression attack

(l) Salt and pepper attack

(m) Scaling attack

(n) Sharpening attack

Figure 22. (a) original image, (b) watermarked image, (c-n) watermarked image with different attacks applied to it.

To evaluate robustness properties of proposed method, various attacks like additive white Gaussian noise, blurring, contrast enhancement, cropping, flipping, gamma correction, Gaussian noise, histogram equalization, JPEG compression, salt and pepper noise, scaling and sharpening were used on watermarked image. Extracted watermark image was evaluated using correlation coefficient metric. It evaluates similarity between extracted watermark and original watermark image. For comparison purposes one conventional and the two state-of-the-art techniques were implemented and measured to evaluate proposed methods experimental results. From conducted tests it can be seen that presented colour image

watermarking scheme outperforms other novel methods. Experimental results show that proposed algorithm performs extremely well when histogram equalization, blurring, contrast enhancement, sharpening and scaling attacks are applied on watermarked image.

Table 12 shows comparison between one conventional and two state-of-the-art algorithms when "Lena" is used as host image and "Cameraman" as watermark image. Table 12 points out that proposed algorithm has significantly better correlation coefficient results with histogram equalization, cropping, blurring, contrast enhancement, Gaussian noise, sharpening and scaling attacks. Correlation coefficient result is slightly better with additive white Gaussian noise attack. Proposed algorithm shows slightly worse correlation coefficient results than other compared methods with JPEG compression, gamma correction and salt and pepper noise attacks.

Table 12. Correlation coefficient values of "Lena" as host image watermarked with "Signature".

Attack	LSB method	Lai & Tsai method [45]	Mary Agoyi *et al.* method [22]	Proposed RGB method
Flipping	0.3441	1	1	1
Histogram equalization	-0.0486	0.8869	0.8713	0.9846
Cropping	0.6417	0.8426	0.8893	0.9427
JPEG	-0.0069	0.8968	0.8837	0.8368
Blurring	0.1412	0.6993	-0.6327	0.9712
Contrast enhancement	0.0653	0.9039	0.8943	0.9947
Salt and pepper noise	0.9871	0.8947	0.5932	0.9659
Gaussian noise	0.0015	0.8330	0.5237	0.9015
Sharpening	0.0745	0.9307	0.8697	0.9571
Gamma correction	-0.1077	0.9302	0.7609	0.7147
Scaling	0.1831	0.7324	-0.6313	0.9673
AWGN	-0.0015	0.7940	0.4932	0.7976

Table 13 shows correlation coefficient values comparison between proposed colour image watermarking method and other state-of-the-art and conventional watermarking techniques, when "Peppers" is used as host image and "UT logo" as watermark. This table shows that proposed colour watermarking method has significantly higher robustness characteristics when blurring, scaling and additive white Gaussian noise attacks are applied on watermarked image. Proposed method shows slightly better robustness results when histogram equalization, cropping, salt and pepper noise, Gaussian noise and sharpening attacks are used. Proposed method shows marginally lower correlation coefficient results when flipping, JPEG compression, contrast enhancement and gamma correction attacks are used.

Table 13. Correlation coefficient values of "Peppers" as host image watermarked with "UT logo".

Attack	LSB method	Lai & Tsai method [45]	Mary Agoyi *et al.* method [22]	Proposed RGB method
Flipping	0.1341	1,0000	1,0000	0.7330
Histogram equalization	-0.0120	0.9698	0.9811	0.9874
Cropping	0.5270	0.9802	0.7750	0.9956
JPEG	0.0167	0.9888	0.5401	0.7279
Blurring	0.0001	0.6430	-0.958	0.8963
Contrast enhancement	-0.0050	0.9804	0.9880	0.9746
Salt and pepper noise	0.8984	0.9528	0.9235	0.9771
Gaussian noise	-0.0032	0.9190	0.8676	0.9234
Sharpening	0.0077	0.9649	0.9768	1,0000
Gamma correction	-0.0536	0.994	0.9850	0.9542
Scaling	0.0072	0.7399	-0.9564	0.8523
AWGN	-0.0145	0.8965	0.8794	0.9578

Figure 23 shows visual results of extracted black and white watermarks when host image is "Lena" and watermark image is "Cameraman". This figure shows that extracted watermarks have good visual quality, they are easily distinguishable and proposed algorithm is able to retrieve watermark after several signal processing attacks.

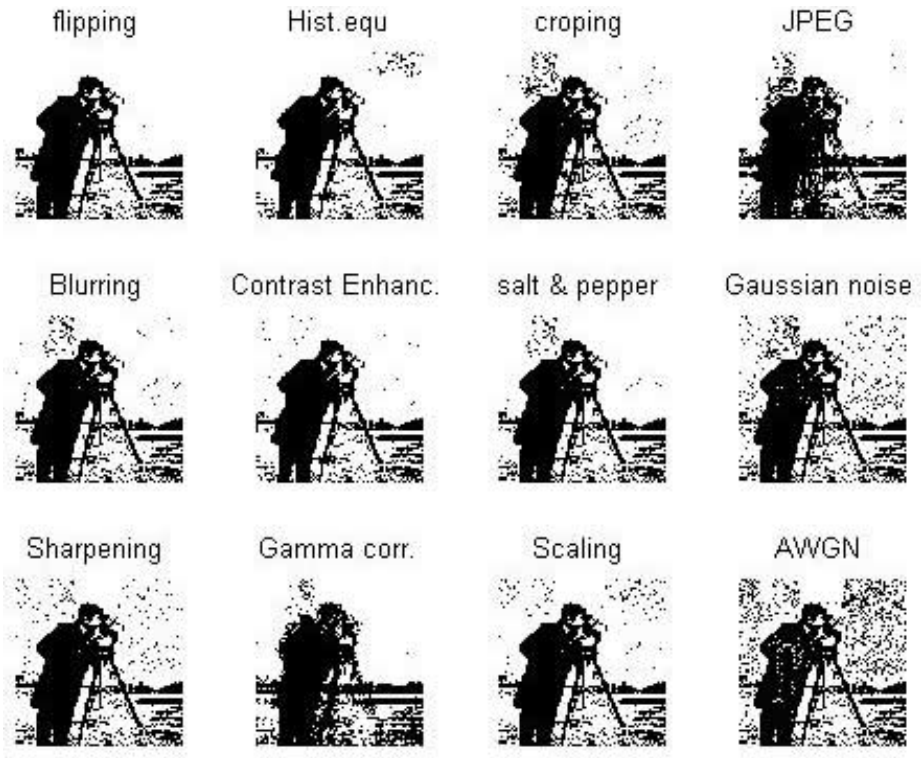


Figure 23. Black and white extracted watermarks from colour image.

6. Conclusion and Future Work

The aim of this thesis was to develop imperceptible and robust watermarking algorithm, what would perform better than other state-of-the-art watermarking methods. The key strength part of developed algorithms is using entropy to determine where to embed watermark in cover image. Entropy of host image shows what image parts are more complex and contain more information. Majority of image processing attacks change high-entropy parts of an image. Embedding watermark into low-entropy parts makes watermarked image more robust against common image processing attacks. Proposed algorithms take advantage of other signal processing methods like SVD, DWT and chirp z-transform.

During presented work one colour and two grayscale image watermarking algorithms were developed and analysed. First grayscale algorithm used QR decomposition and the second used LU decomposition. Experimental results showed that grayscale algorithm what used LU decomposition had better experimental results, thus it was used to develop colour image watermarking scheme.

As a result of this thesis, novel grayscale and colour image watermarking algorithms were developed. Many experiments were conducted and proposed algorithms were compared with other state-of-the-art algorithms. Experimental results showed that all proposed algorithms outperform other conventional and cutting edge watermarking methods. Comparison tables showed that proposed methods have higher overall metric values. Proposed algorithms are robust and they produce watermarked images what have no perceptual difference with original images. Extracted watermarks are easily distinguishable and they have good visual quality.

This work offers many opportunities for future work. Developed techniques are non-blind algorithms, meaning that they need original image, watermark and secret key to extract the watermark. Although there are many application areas for non-blind watermarking algorithms, like content validation, broadcast monitoring, ownership verification and others, proposed algorithms can be used to develop good entropy based blind watermarking schemas. Videos consist of sequence of frames, other logical continuation for developed colour image watermarking algorithm could be to use it in video watermarking.

7. References

[1] S. Landau and M. R. Stytz, "Overview of cyber security: a crisis of prioritization," *Security & Privacy, IEEE,* vol. 3, no. 3, pp. 9-11, 2005.

[2] A. B. Hamida, M. Koubaa, C. B. Amar and H. Nicolas, "Hierarchical traceability of multimedia documents," in *IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, Paris, 2011.

[3] W. Chung and J. Paynter, "Privacy issues on the Internet," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, Hawaii, 2002.

[4] M. Campidoglio, F. Campidoglio and F. Landolfi, "The Copyright Protection Problem: Challenges and Suggestions," in *Fourth International Conference on Internet and Web Applications and Services*, Venice/Mestre, 2009.

[5] G. Bleumer, "Watermarking," in *Encyclopedia of Cryptography and Security*, Springer US, 2011, pp. 1365-1366 .

[6] Y. Q. Shi, A. T. S. Ho, M. Barni and H. J. Kim, Digital Watermarking, Guildford, UK: Springer Verlag, 2009.

[7] M. A. Dorairangaswamy and B. Padhmavathi, "An effective blind watermarking scheme for protecting rightful ownership of digital images," in *2009 IEEE Region 10 Conference TENCON*, Singapore, 2009.

[8] F. Bartolini, G. Bini, V. Cappellini, A. Fringuelli, G. Meucci, A. Piva and M. Barni, "Enforcement of copyright laws for multimedia through blind, detectable, reversible watermarking," in *IEEE International Conference on Multimedia Computing and Systems*, Florence, 1999.

[9] T. Minamoto and R. Ohura, "A Non-blind Digital Image Watermarking Method Based on the Dual-tree Complex Discrete Wavelet Transform and Interval Arithmetic," in *Ninth International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, NV, 2012.

[10] K. Tanaka, Y. Nakamura and K. Matsui, "Embedding secret information into a dithered multi-level image," in *MILCOM '90, Conference Record, A New Era. 1990 IEEE Military Communications Conference*, Monterey, CA, 1990.

[11] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079 - 1107 , 1999.

[12] R. J. Anderson, Information Hiding, First International Workshop, Cambridge, U.K.: Springer, 1996.

[13] C. Ingemar, M. Miller, J. Bloom, J. Fridrich and T. Kalker, Digital Watermarking and Steganography, Morgan Kaufmann, 2007.

[14] M. D. Swanson, M. Kobayashi and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proceedings of the IEEE,* vol. 86, no. 6, pp. 1064 - 1087 , 1998.

[15] J. Dittmann, A. Mukherjee and M. Steinebach, "Media-independent watermarking classification and the need for combining digital video and audio watermarking for media authentication," in *International Conference on Information Technology: Coding and Computing*, Las Vegas, 2000.

[16] L. Liu and X. Li, "Watermarking Protocol for Broadcast Monitoring," in *International Conference on E-Business and E-Government (ICEE),* Guangzhou, 2010.

[17] I. J. Cox, M. L. Miller and J. A. Bloom, "Watermarking applications and their properties," in *International Conference on Information Technology: Coding and Computing*, Las Vegas, 2000.

[18] V. M. Potdar, S. Han and E. Chang, "A Survey of Digital Image Watermarking Techniques," in *3rd IEEE International Conference on Industrial Informatics*, Perth, 2005.

[19] Y. Yusof and O. O. Khalifa, "Imperceptibility and Robustness Analysis of DWT-based Digital Image," in *International Conference on Computer and Communication Engineering*, Kuala Lumpur , 2008.

[20] L. Perez-Freire, P. Comesana, J. R. Troncoso-Pastoriza and F. Perez-Gonzalez, "Watermarking Security: A Survey," in *Transactions on Data Hiding and Multimedia Security,* Springer Berlin Heidelberg, 2006, pp. 41-72.

[21] F. Yaghmaee and M. Jamzad, "Computing watermark capacity in images according to their quad tree," in *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology*, Athens, 2005.

[22] M. Agoyi, E. Çelebi and G. Anbarjafari, "A watermarking algorithm based on chirp z-transform, discrete wavelet transform, and singular value decomposition," *Signal, Image and Video Processing,* vol. 9, no. 3, pp. 735-745 , 2014.

[23] A. K. Singh, N. Sharma, M. Dave and A. Mohan, "A Novel Technique for Digital Image Watermarking in Spatial Domain," in *2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC),* Solan, 2012.

[24] H. H. Larijani and G. R. Rad, "A New Spatial Domain Algorithm for Gray Scale Images Watermarking," in *International Conference on Computer and Communication Engineering*, Kuala Lumpur, Malaysia, 2008.

[25] M. Grgic, M. Ravnjak and B. Zovko-Cihlar, “Filter comparison in wavelet transform of still images,” in *Proceedings of the IEEE International Symposium on Industrial Electronics*, Bled, 1999.

[26] B. Furht, “Discrete Wavelet Transform (DWT),” in *Encyclopedia of Multimedia*, Springer US, 2008, p. 188.

[27] D. Dejey and R. Rajesh, “Robust discrete wavelet-fan beam transforms-based colour image watermarking,” *IET Image Processing*, vol. 5, no. 4, pp. 315-322, 2011.

[28] N. Ahmed, T. Natarajan and K. R. Rao, “Discrete Cosine Transform,” *Computers, IEEE Transactions on*, Vols. C-23, no. 1, pp. 90-93, 1974.

[29] R. Dubolia, R. Singh, S. S. Bhadoria and R. Gupta, “Digital Image Watermarking by Using Discrete Wavelet Transform and Discrete Cosine Transform and Comparison Based on PSNR,” in *International Conference on Communication Systems and Network Technologies (CSNT)*, Katra, Jammu, 2011.

[30] M. Iwakiri and M. Iwakiri, “Fundamental Incomplete Cryptography Method to Digital Rights Management Based on JPEG Lossy Compression,” in *IEEE 26th International Conference on Advanced Information Networking and Applications (AINA)*, Fukuoka, 2012.

[31] A. Shaamala, S. M. Abdullah and A. A. Manaf, “Study of the effect DCT and DWT domains on the imperceptibility and robustness of Genetic watermarking,” *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 5, pp. 220-225, 2011.

[32] I. Amidror, Mastering the Discrete Fourier Transform in One, Two or Several Dimensions, London: Springer, 2013.

[33] K. Bhandari, S. K. Mitra and A. Jadhav, “A Hybrid Approach to Digital Image Watermarking Using Singular Value Decomposition and Spread Spectrum,” in *Pattern Recognition and Machine Intelligence*, Springer Berlin Heidelberg, 2005, pp. 447-452.

[34] A. Mansouri, A. M. Aznaveh, F. Torkamani-Azar and J. A. Jahanshahi, “Image quality assessment using the singular value decomposition theorem,” *Optical Review* , vol. 16, no. 2, pp. 49-53, 2009.

[35] D. Kuobin, “Singular Value Decomposition Watermarking Method for Medical Image,” in *International Conference on Intelligence Science and Information Engineering (ISIE)*, Wuhan, 2011.

[36] L. Zhi and C. Xiao-Wei, “Self-Adaptive Video Watermarking Based on the Motion Characteristic Detection and the Model of Entropy,” in *IIHMSP '08 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Harbin, 2008.

[37] Q. Yang, Y. Zhang, C. Yang and W. Li, “Information Entropy Used in Digital Watermarking,” in *Symposium on Photonics and Optoelectronics (SOPO)*, Shanghai, 2012.

[38] Y. Zheng, C. Qi and G. Wang, “A New Image Pre-Processing for Improved Performance of Entropy Coding,” in *Chinese Conference on Pattern Recognition (CCPR)*, Chongqing, 2010.

[39] G. Wu, Y. Dou, J. Sun and G. D. Peterson, “A High Performance and Memory Efficient LU Decomposer on FPGAs,” *IEEE Transactions on Computers*, vol. 61, no. 3, pp. 366 - 378 , 2012.

[40] J. Onur, H. G. Ilk and E. Elbasi, “A secure and robust watermarking algorithm based on the combination of DWT, SVD, and LU decomposition with Arnold's Cat Map approach,” in *8th International Conference on Electrical and Electronics Engineering (ELECO)*, Bursa, 2013.

[41] S.-C. Han and Z.-N. Zhang, “A novel zero-watermark algorithm based on LU decomposition in NSST domain,” in *IEEE 11th International Conference on Signal Processing (ICSP)*, Beijing, 2012.

[42] M. K. Jaiswal and N. Chandrachoodan, “FPGA-Based High-Performance and Scalable Block LU Decomposition Architecture,” *IEEE Transactions on Computers*, vol. 61, no. 1, pp. 60-72, 2011.

[43] G. Kaur and K. Kaur, “Image Watermarking Using LSB (Least Significant Bit),” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 4, pp. 858-861, 2013.

[44] B. Pandhwal and D. S. Chaudhari, “An Overview of Digital Watermarking Techniques,” *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 3, no. 1, pp. 416-420, 2013.

[45] C.-C. Lai and C.-C. Tsai, “Digital Image Watermarking Using Discrete Wavelet Transform and Singular Value Decomposition,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 11, pp. 3060-3063, 2010.

[46] E. Ganic and A. M. Eskicioglu, “Robust DWT-SVD domain image watermarking: Embedding data in all frequencies,” in *Proceedings of the 2004 workshop on Multimedia and security*, New York, 2004.

[47] R. Liu and T. Tan, “An SVD-based watermarking scheme for protecting rightful ownership,” *IEEE Transactions on Multimedia*, vol. 4, no. 1, pp. 121 - 128 , 2002.

[48] S. Wang, D. Zheng, J. Zhao, W. J. Tam and F. Speranza, “Adaptive Watermarking and Tree Structure Based Image Quality Estimation,” *IEEE Transactions on Multimedia*, vol.

16, no. 2, pp. 311 - 325, 2013.

[49] J. Y. Zheng, D. H. Liang, J. Z. Liang and M. Jin, “A DCT-BASED Digital Watermarking Algorithm for Image,” in *International Conference on Industrial Control and Electronics Engineering (ICICEE)*, Xi'an, 2012.

Appendix

I. License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Lauri Laur** (date of birth: 25.08.1990),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

    1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

    1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright, **Entropy Based Robust Watermarking Algorithm**, supervised by Assoc. Prof. Gholamreza Anbarjafari and Asst. Prof. Mary Agoyi.

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **20.05.2015**