

# Framework for robust blind image watermarking based on classification of attacks

M. Heidari<sup>1,2</sup> · S. Samavi<sup>2,3</sup> · S. M. R. Soroushmehr<sup>4</sup> ·  
S. Shirani<sup>5</sup> · N. Karimi<sup>2</sup> · K. Najarian<sup>3,4,6</sup>

Received: 21 March 2016 / Revised: 17 September 2016 / Accepted: 10 November 2016 /  
Published online: 18 November 2016  
© Springer Science+Business Media New York 2016

**Abstract** With the widespread internet usage, digital contents are easily distributed throughout the world. To eliminate concerns of producers and owners of digital contents, watermarking techniques are extensively being used. Robustness against intentional and unintentional attacks is a major quality of watermarking systems. Since different attacks tend to target different parts of the frequency spectrum, in this paper we propose a framework for blind watermarking which determines the type of attack that the image has gone through before extracting the watermark. Within this framework, we propose an attack classification method to identify the region of the frequency spectrum that is less damaged. The watermark which is redundantly spread throughout the spectrum can be extracted from the less damaged regions. Experimental results show functionality of the framework by producing better results in comparison with well-known blind watermarking techniques.

**Keywords** Watermarking · Robustness · General framework · Invisibility

---

✉ S. Samavi  
samavi@mcmaster.ca

<sup>1</sup> Department of Electrical Engineering, Sharif University of Technology, Tehran 11365-11155, Iran

<sup>2</sup> Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran

<sup>3</sup> Department of Emergency Medicine, University of Michigan, Ann Arbor 48109, USA

<sup>4</sup> Michigan Center for Integrative Research in Critical Care, University of Michigan, Ann Arbor 48109, USA

<sup>5</sup> Department of Electrical and Computer Engineering, McMaster University, L8S4L8, Hamilton, ON, Canada

<sup>6</sup> Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor 48109, USA

## 1 Introduction

With the development of computers and internet, more and more people tend to use them for their day-to-day communication needs. Sharing of digital media products in social networks has become increasingly conventional. Therefore, digital media products are subjects of piracy and plagiarism, and protecting them against illegal copyright is a major concern for many digital forensic experts. We can take advantage of digital watermarking as a reliable way to embed a message into multimedia documents to resolve copyright ownership concerns.

Robustness, invisibility, and capacity are three major characteristics of a watermarking algorithm [16]. A watermarking system is *invisible* if the original cover image and the watermarked image are perceptually indistinguishable. The ability to withstand against attacks, such as additive noise, filtering, compression, cropping, etc., is referred to as the *robustness* of the algorithm. Robustness allows correct extraction of the watermark after the image has gone through attacks. The third characteristic is *capacity*, which is of lower importance as compared to invisibility and robustness. The capacity of a watermarking system is the maximum amount of message that we can insert in the cover image without noticeable loss in the quality of the image [19].

We can use either spatial domain or transform domain as the watermarking space for our scheme [21]. Spatial domain watermarking methods usually are more vulnerable to signal processing attacks. On the other hand, embedding in transform domain gives more capacity and results in more robustness [9].

Burgett et al. presented the first efficient watermarking in the discrete cosine transform (DCT) domain [5]. In their algorithm, the image is partitioned into  $8 \times 8$  blocks for which the DCT is computed. In each block, a pair of coefficients is selected. To embed a bit, the coefficients are switched around so that their positional difference is either positive or negative, depending on the bit value. Another watermarking method is presented in [3] that is based on discrete wavelet transform (DWT) and DCT. Their experimental results show that this method satisfies perceptual concerns of a watermarking system, but it is not robust enough against strong noise attacks. The method of [14] models the watermarking channel as a generalized channel with fading and nonzero mean additive noise. They propose an optimized watermark extraction scheme by using an adaptive extractor for their quantization-based watermarking. In [1], 4 levels of DWT are selected as the domain of watermarking. This is done to make the method robust while keeping the watermark invisible. Their method is adaptive based on human visual system (HVS). In [12] wavelet is the domain of choice for embedding of the watermark. They use eight of the approximation coefficients in every image block to construct two line segments in the special domain. The angle between the two line segments is changed to embed data. Watermarking in low-frequency coefficients of host image based on a sample projection approach is a technique that is analyzed in [2]. It uses the slope of a line segment in the 2-D space, generated by four samples of the approximation coefficients of the image blocks, for embedding purpose. It uses a maximum likelihood (ML) decoder for data extraction and the results show its advantage to the other similar systems. The algorithm in [20] inserts the watermark in compressed JPEG2000 and encrypted images. For encryption step, this method takes advantage of stream cypher system. In [27] a quantized watermarking method based on feature modulation is developed. Feature signal in this method is the normalized correlation (NC) formed by the host signal and a certain random signal. A structured codebook is designed to replace calculated NC. The watermark signal is a combination of the cover signal and some random signal. Their goal is to minimize the embedding

distortion. In this method the embedding distortion is inversely related to the quantization step and it is proportional to the variance of the cover image. In [17] to devise an algorithm robust against geometric attacks, they use wavelet domain. Features are extracted from wavelet coefficients and are used as synchronization marks for embedding and extraction purposes. Circular regions around the feature points are selected and the selected region is normalized. DCT of the selected area is used for embedding purposes. In [15] they use singular value decomposition (SVD) for the embedding domain. They apply SVD to image blocks and form a matrix from the first singular values of all blocks. DCT of this matrix provides means of modifications for embedding of the data. Different embedding and extraction methods have been considered in the literature. For example, in [6, 7] genetic algorithm (GA) has been used for watermarking purposes. Also worth mentioning is that one of the major applications of watermarking is forgery detection. In [10, 26] watermarking is used for tamper detection in different multimedia environments. In [11] a watermarking method based on contourlet transform (CT) is proposed. They embed the watermark in DCT coefficients of CT's detail blocks. They embed with higher strength factor in complex blocks. This is done to achieve higher robustness. Also, weaker embedding is performed in smooth blocks to satisfy imperceptibility. Edge concentration and entropy of blocks are used to determine complexity of each detail block. They also take advantage of redundant embedding and voting mechanism to further increase the robustness of their method.

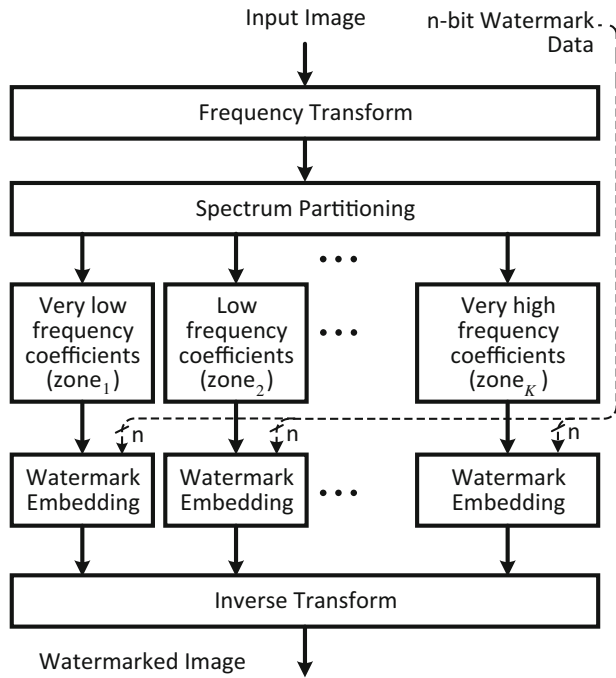
There are various noise attacks and image enhancement filters, and each one has its own destructive effects on a specific frequency-band of the image. A broad look over the investigated methods shows that each watermarking method could resist against a number of these attacks. Some methods insert messages into low frequency components and they are robust against noise attacks. But such methods lose visual quality of the image and they are not immune against sharpening and high pass filter attacks. Some other techniques embed the watermark in mid-band coefficients to improve visual quality, but they are just robust to moderate attacks. A method which uses both low and high frequency components is useful to be robust against a wide range of attacks.

In this paper we propose a framework for image watermarking to simultaneously satisfy invisibility and robustness criteria. We noticed that some attacks tend to destroy high frequency components more and some attacks tend to change low frequency components. Hence, we propose a framework that embeds the watermark redundantly in different parts of the frequency spectrum. Block diagram of the embedding part of the proposed framework is shown in Fig. 1.

When it comes to extraction of the watermark we use a classifier which has already been trained to identify what part of the frequency spectrum of the watermarked image has been least damaged. The extraction part of the framework is shown in Fig. 2. Classification is done using a set of features that are extracted from the frequency domain of the watermarked image. Then the extraction is performed on the data that is less affected by the attack. Hence, the extraction part is completely blind and it neither needs the original image nor it needs the original watermark.

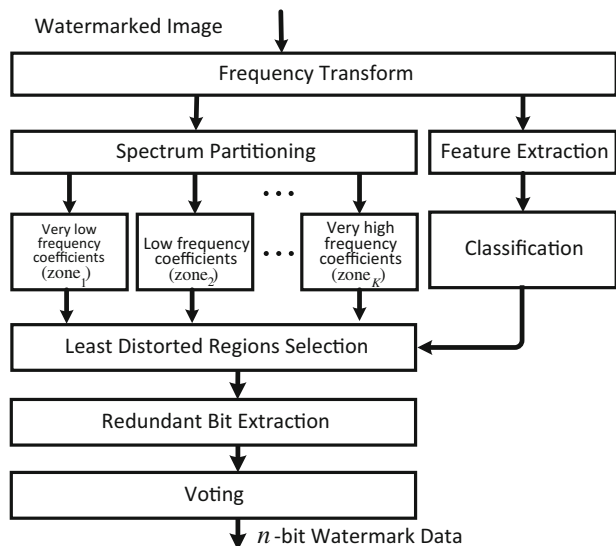
To test our proposed framework we present an embedding method and then we present a classifier and an extraction method. But the main goal of this research is the above mentioned framework. Different embedding methods and classifiers could be used in this framework. As an example, we use DCT as the embedding domain. We are not emphasizing on the use of this domain. Instead of DCT, we could use DWT, DFT (discrete Fourier transform), or other frequency transform domains. As long as the transform domain provides a wide range of the

**Fig. 1** Block diagram of the embedding part of the proposed framework



frequency spectrum it would be suitable to be used in our framework. In the transform stage, instead of blocking the image into  $8 \times 8$  sub-blocks and calculating DCT of each sub-block separately, we consider the DCT of the whole image. This helps eliminating blockiness artifacts. In the extraction phase, as an example, we use a support vector machine (SVM) classifier to show which frequency region has been more affected by the attack. Other

**Fig. 2** Block diagram of the extraction part of the proposed framework



classifiers, such as decision tree (DT), k-nearest neighbor (k-NN), logistic regression (LogR), or naïve Bayes (NB), could be used instead of SVM. Finally, using a voting technique on all extracted bits, the watermark is reconstructed. Hence, the novelty of our work is not the DCT embedding method but the proposed framework is the main novelty of this paper. We compare our results with the state-of-the-art methods.

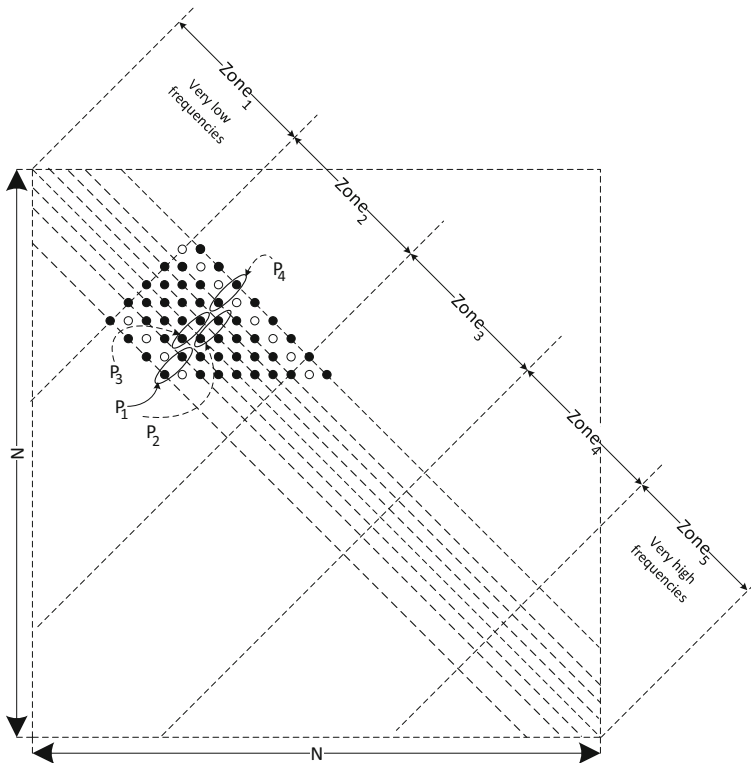
The rest of the paper is organized as follows. Details of proposed watermark embedding and extraction techniques are presented in Section 2 and Section 3 respectively. Section 4 provides results of the proposed method. Finally, we conclude our method in Section 5.

## 2 Watermark embedding

In this section, we explain our embedding algorithm within the proposed framework. The framework requires an initial frequency transform of the image. As an example we choose DCT. First the DCT of the whole image is calculated. Jiang and Feng, in [13] mathematically show a linear relationship between DCT coefficients of a block and the DCT coefficients of its sub-blocks. Therefore, each DCT coefficient, when it is calculated for the whole image, is a combination of DCT coefficients of its  $8 \times 8$  sub-blocks. Hence, if we change a coefficient, for data embedding purposes, its effects are diffused into the DCT coefficients of its sub-blocks. This causes a small change in sub-block coefficients. Thus, global method makes smaller distortion on local blocks. Furthermore, when the DCT of the whole image is considered, modification effects are diffused on the entire image. We expect that distortions are negligible as compared to the blockiness effects that occur in block-based algorithms.

Embedding is performed by repositioning or altering some of the DCT coefficients. This is inspired by method of [5]. If we use a zigzag-scanning technique to scan coefficients from the lowest to the highest frequency, every two consecutive components have high correlation with each other. For watermarking, to cause minimum distortion in the image, changes in DCT components should be very small. We select pairs of consecutive coefficients close to the main diagonal. To embed a bit, the coefficients are switched around so that their positional difference is either positive or negative, depending on whether we want to embed a 1 or a 0. The embedding region that we consider in the DCT matrix is alongside the main diagonal of the DCT matrix from top left to the bottom right corner of the matrix. The main diagonal and its seven surrounding diagonals are considered. This gives us 4 pairs of consecutive coefficients around the coordinates  $(i, i)$ . Hence, the pair  $P_1$  consists of two DCT coefficients of  $D(i+2, i-2)$  and  $D(i+1, i-1)$ . With the same token, the other 3 pairs are:  $P_2: D(i, i) \& D(i-1, i+1)$ ,  $P_3: D(i, i-1) \& D(i-1, i)$ , and  $P_4: D(i-2, i+1) \& D(i-3, i+2)$ . For ease of reference, in every pair of consecutive coefficients, we call the first coefficient as  $D_1$  and the second one as  $D_2$ . Characteristic of these coefficients, which are close to the main diagonal, is that their horizontal and vertical frequencies are almost equal. Hence, when we switch around the two coefficients for data insertion, it has the same spatial effect in both directions. This narrow strip of eight diagonal strings of coefficients is shown in Fig. 3.

Based on the framework of Fig. 1 we need to partition the frequency spectrum into  $K$  zones and redundantly embed in some of these zones. In the sample embedding algorithm, as shown in Fig. 3, we choose  $K=5$ . Other number of zones could have been chosen too. Hence, we partition the embedding region, from very low to very high frequencies, into 5 zones. Embedding in very low frequency coefficients results in high robustness against many of the attacks but it has strong negative effect on the quality of the image. Also, embedding in



**Fig. 3** DCT coefficient matrix. Embedding region is a strip consisting main diagonal and 7 surrounding diagonals. Example of 4 coefficients pairs and partitioning of frequency spectrum into 5 zones are shown. Solid dots are coefficients used for watermarking, and hallow circles are coefficients not involved in the embedding

very high frequency components has low effects on invisibility but it would be very vulnerable to attacks. We do not have to, but we choose not to embed in the very low frequency coefficients ( $Zone_1$ ) and very high frequency coefficients ( $Zone_5$ ). Hence, in our sample implementation of the framework, embedding is done only in  $Zone_2$ ,  $Zone_3$ , and  $Zone_4$ .

Every two neighboring coefficients have the highest correlation. For each bit of watermark, the coder selects two of such DCT coefficients in the mentioned region of the matrix. As shown in the pseudo code of Fig. 4, depending on which of the two coefficients is bigger, a bit could be embedded in that pair.

To embed a bit redundantly, every watermark bit is repeatedly embedded in four pairs of coefficients. Examples of such 4 coefficient pairs are shown in Fig. 3 as  $P_1$ ,  $P_2$ ,  $P_3$ , and  $P_4$ .

**Fig. 4** Embedding procedure for each zone.  $w(m)$  is the  $m$ th bit of a watermark string of  $L$  bits. Each bit is embedded in 4 coefficient pairs

```

FOR m = 1 to L
  FOR p = 1 to 4 //four coefficient pairs
    IF ( $w(m) = 0$  AND  $D_1 \geq D_2$ ) OR
       ( $w(m) = 1$  AND  $D_1 < D_2$ )
      swap values of coefficients
    ELSE
      do nothing
    ENDIF
  ENDFOR
ENDFOR

```

Suppose that we have a watermark string with  $L$  bits and let  $w(m)$  be the  $m$ th bit of this string. It is shown in the pseudo code of Fig. 4 that if the watermark bit is zero, then the two coefficients should be placed such that the smaller of the two be located in the position with smaller column number. This method of embedding makes the algorithm blind. When it comes to extracting the embedded bit, we do not need the original cover image.

In this sample implementation, when embedding the watermark in a zone, we are embedding each bit 4 times. This is done in 3 zones, hence, embedding of the watermark is performed with an overall redundancy of 12.

### 3 Extraction procedure

At the receiver end, we need to extract the embedded watermark from the watermarked image which could have been attacked. The received image is transformed and by analyzing coefficient pairs we should be able to extract the embedded watermark. Coefficients of the three mentioned zones are considered. In the DCT matrix, frequencies of the mentioned zones are such that  $frequency_2 < frequency_3 < frequency_4$ . Some attacks, such as high-pass filters, affect coefficients of  $Zone_2$  more than they affect  $Zone_4$  coefficients. Other attacks, such as low-pass filters, may have higher effects on  $Zone_4$  coefficients. Therefore, based on the framework, before performing a voting procedure on the extracted bits, we need to discover which coefficients are affected more. Then we can have a more reliable voting by using coefficients that have been less modified.

Features sensitive to attacks are extracted. They are fed into a trained classifier to label the watermarked image. This label shows which part of the frequency spectrum has been more damaged in the watermarked image due to attacks. In our sample implementation we have two classes, depending on whether high frequency coefficients have been affected more or low frequency ones. Next, the decoder is fed with the discovered class label of the watermarked image. Then, the distorted embedding zone is neglected and the decoder extracts eight versions of the watermark from the other two less distorted zones. The voting will be performed and the final watermark is reconstructed.

#### 3.1 Feature extraction

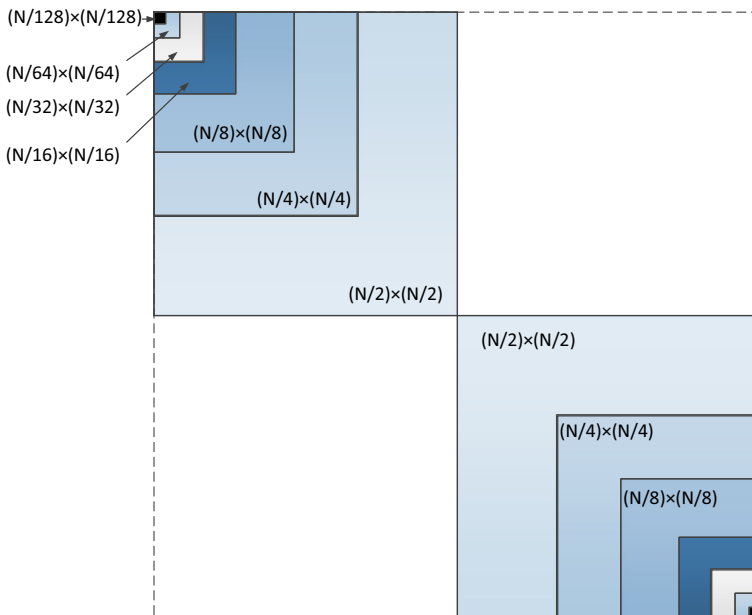
Based on Fig. 2, the proposed framework requires extraction of a set of features that are sensitive to attacks. Here we propose a set sample features. Some attacks cause serious changes in high frequency coefficients. Such attacks include median, Gaussian and average filters; some noises like Gaussian and speckle; data compression tools like JPEG compressions, and geometric function of resize. We place images affected by such attacks in class **High**. Furthermore, all high-pass filters and those that sharpen the image, destroy watermarks that are embedded in low frequency coefficients. Hence, they are in class **Low**. Some attacks like; cut and crop, depending on their action to the image can be in class **High** or **Low**. If the cropped area is smooth, this would ruin the watermark that is embedded in  $zone_2$  and such image would be of class **Low**. Otherwise, if the attack crops a crowded area the resulted image would fall in class **High**.

It is shown in [4] that when the noise added to the image is not correlated with the image, which is usually the case, DCT could reveal the presence of the noise. Hence, to study changes

in the DCT matrix we perform statistical analysis on groups of coefficients that are in our region of interest. We will partition the matrix into a number of overlapped blocks as shown in Fig. 5. We first partition the matrix into 4 quadrants. The top-left and the bottom-right quadrants are of our interest. Statistical analysis for feature extraction, are performed on all of coefficients of these two quadrants. Then at the second level of partitioning, each of the mentioned quadrants is partitioned into sub-blocks and coefficients of one sub-block are used for feature extraction. For an  $N \times N$  image, this partitioning will be done for  $\log_2(N) - 2$  times resulting two groups, each having  $G$  sub-blocks, where  $G = \log_2(N) - 2$ . For a  $2^Q \times 2^Q$  image, sub-blocks are of the size  $2^q \times 2^q$ , where  $q = Q - 1, \dots, 3, 2$ . Figure 5 shows an example of the proposed partitioning. In the example of Fig. 5 we have 7 sub-blocks in the top-left quadrant and the same number in the bottom-right. Hence, for such partitioning we have 14 sets of coefficients for our statistical analysis. In the followings we list features to reveal distributions in DCT coefficients within our region of interest.

**Entropy** Entropy of every sub-block is calculated. This is done by generating a histogram of the coefficients of a sub-block. Then the probability of occurrence of each bin of the histogram is calculated. The entropy of a sub-block is expected to increase when noise is added to the image or when the image is enhanced.

**Variance and energy** Let  $D_i$  be the value of  $i$ th coefficient in a sub-block which has  $g$  coefficients, and  $\mu$  is the mean value of those coefficients. Then the variance of the coefficients of that block would be expressed by (1). In (1) we have changed the original variance equation by adding a 1 to better show the effect of values that are smaller than 1. The energy of the components is expressed by (2).



**Fig. 5** Topology for feature extraction from DCT matrix using overlapped block-partitioning



$$\text{Var} = \frac{1}{g-1} \sum_{i=1}^g (|D_i - \mu| + 1)^2; \mu = \frac{1}{g} \sum_{i=1}^g D_i \quad (1)$$

$$\text{Energy} = \sum_{i=1}^g (D_i)^2 \quad (2)$$

**Sum of singular values** In [22] it is proved that changes in the spatial domain of an image has a direct effect on its DCT coefficients distribution. Singular value decomposition (SVD) is a well-suited transform that extracts the rank of a matrix. Hence, we use SVD of the  $j$ th DCT sub-block for feature extraction. The  $i$ th non-zero singular value of the  $j$ th sub-block is  $\sigma_{ij}$ . Let  $M$  be the number of non-zero singular values and  $G$  be the number of sub-blocks. As shown in (3), the sum of log of inverse of singular values, of the  $j$ th sub-block, is  $C_j$  and will be used as a feature.

$$C_j = \frac{1}{M} \sum_{i=1}^M \log(\sigma_{ij}^{-1}) \quad , j = 1, 2, \dots, G; \sigma_{ij}^{-1} \neq 0 \quad (3)$$

**Geometric mean of singular values** Using (4), multiplication of singular values in the form of geometric mean is also used as another feature.

$$\Gamma_j = \left( \prod_{i=1}^M \sigma_{ij}^{-1} \right)^{\frac{1}{M}} \quad , j = 1, 2, \dots, G; \sigma_{ij}^{-1} \neq 0 \quad (4)$$

**Moments of distributions** It is shown that analysis of moments of a distribution, for a sequence of independent elements, could reveal changes that have occurred in that distribution [24]. The  $n$ th moment of distribution for the  $j$ th sub-block's singular values is shown in (5).

$$\begin{aligned} \mu_j &= \sum_{i=1}^M \Pr(\sigma_{ij}) \sigma_{ij} \\ m_n^j &= \sum_{i=1}^M \Pr(\sigma_{ij}) (\sigma_{ij} - \mu_j)^n, \\ j &= 1, 2, \dots, G; \quad n = 1, 2, 3, 4 \end{aligned} \quad (5)$$

where  $\mu_j$  is the mean value of the singular values in the  $j$ th sub-block and  $\Pr(\sigma_{ij})$  is the probability of occurrence of  $i$ th singular value of the  $j$ th sub-block. As mentioned in (5), we use the first four moments as the last set of features. We see that we have nine features for each of the fourteen sub-blocks, when we are dealing for example with  $512 \times 512$  images. This would form a 126 dimensional feature vector which is sensitive to different attacks.

### 3.2 Classification

The proposed framework is not bound to any specific classifier. As a sample, the extracted 126 dimensional feature vector is used in an SVM learning machine to classify the type of attack.

Every image that is used for training is labeled as *High* or *Low*, based on the known attack that the image has gone through. Then, after the training procedure, the SVM is fed with an input image and the feature vector of that image and is able to assign one of the two mentioned classes to that image.

To test the performance of the trained SVM, we use images from CorelDraw image dataset [8]. We cropped images to be  $512 \times 512$  and converted them to grayscale. Since JPEG is a popular format, we converted a group of our images to JPEG format with the quality factor of 85%. Then in the encoder, a pseudorandom 128-bit watermark was embedded in all of the images. There were 1100 images in the *High* class and 800 images in the *Low* class. From the two classes 1300 images were randomly selected for training and the other 600 images for the test step. This system is implemented by Lib-SVM toolbox with RBF as kernel of the classifier.

For the 600 test images, which were different than the training images, we obtained the following results:

$$\text{False negative (FN)} = 5, \text{ False positive (FP)} = 3,$$

$$\text{True negative (TN)} = 423, \text{ and True positive (TP)} = 169$$

Hence, the accuracy of the system,  $\frac{TP+TN}{TP+TN+FP+FN}$ , turned out to be 98.5%.

### 3.3 Bit extraction and voting

For the extraction step, based on the class of the image, bits are extracted from two of the three zones. If class of image found to be *High* it means that high frequency coefficients are damaged, hence the extraction is performed on *Zone*<sub>2</sub> and *Zone*<sub>3</sub>. On the other hand, if image of classified as *Low* it means that low frequency coefficients have mainly targeted by the attack and the watermark should be extracted from *Zone*<sub>3</sub> and *Zone*<sub>4</sub>. The extraction procedure is reverse of that of the embedding. The difference between two coefficients of a pair determines whether a 1 or a 0 is embedded in that pair. For the purpose of demonstrating the functionality of our framework we picked some implementation choices. When the mentioned difference between two coefficients is larger than a threshold we consider the extracted bit as a *strong-bit*. In the voting process, a strong-bit has 3 times weight as other extracted bits. If this extracted bit is a strong 1 then a weight of +3, and if it is a strong zero, a weight of -3, is assigned to that bit. There are four copies of the watermark in each zone. Hence, we extract eight watermark strings from the two zones. A voting process will give the final extracted watermark.

Effects of many attacks, such as JPEG, are not bounded to only one zone. For example, JPEG basically destroys high frequency coefficients (*Zone*<sub>4</sub>) but when its quality degrades it affects mid-frequencies (*Zone*<sub>3</sub>) too. If we use max rule on all three zones, then altered bits of *Zone*<sub>3</sub> and *Zone*<sub>4</sub> would win. Hence, voting alone could be misleading when two of the three zones are affected. We use SVM to identify *Zone*<sub>4</sub> as being highly affected and block this zone from the voting. We then give higher weight to the less affected *Zone*<sub>2</sub>. Then a reliable voting is expected.

The framework is not restricted to the DCT domain for the embedding purpose. For example, we can use wavelet as the transform domain. If DWT were to be used, we could use two level decomposition of the image. Two copies of the watermark would be embedded in the horizontal and vertical sub-bands of the first level (i.e. *LH*<sub>1</sub> and *HL*<sub>1</sub>). Also, three other copies of the watermark are embedded in the diagonal, horizontal, and vertical sub-bands of the second level (i.e. *HH*<sub>2</sub>, *HL*<sub>2</sub>, and *LH*<sub>2</sub>). The approximate sub-band of the second level (*LL*<sub>2</sub>)

and the diagonal sub-band of the first level ( $HH_1$ ) are not embedded in. Hence, a five-time redundancy is achieved using DWT. Off course, this is just an example to show that the proposed frame work is not bounded with the embedding method or domain.

## 4 Experimental results

Here we want to test the sample watermarking method that is implemented based on the proposed framework. We performed extensive simulations to assess the reliability and performance of the method against several common attacks. We compared our work with state-of-the-art methods of [2, 11, 12, 15, 17]. Hence, experiments were performed on  $512 \times 512$  images. But the framework is good for any size images. Experimenting with  $512 \times 512$  natural images, we chose  $Zone_2$  to begin at coordinates (80, 80) and  $Zone_4$  end at coordinates (463,463). In this section invisibility of our implementations, in terms of visual quality of the watermarked image, is shown. Also, the robustness of the sample implementation is examined against different attacks and results are compared with other existing methods.

### 4.1 Watermark invisibility

As an example for visual quality of the mentioned embedding method, we embedded a 128-bit binary watermark string into four standard images as shown in Fig. 6. The left column of Fig. 6 shows original images and the right column shows the watermarked images and their corresponding Peak Signal to Noise Ratio (PSNR), and the Structural SIMilarity index (SSIM) [25] values.

We also calculate PSNR values of the proposed scheme when applied to 100 images from the USC-SIPI image dataset in [23]. Results are shown in Fig. 7. Average PSNR value for all these 100 images is equal to 44.18 dB. This is an indication of creditable invisibility of the proposed method.

To compare invisibility of the embedding method with other algorithms, we compare PSNR values with schemes in [2, 12, 17]. Images that are used in our experiments are not used for the training of the classifier. The methods used in [2, 12, 17], insert watermark in low frequency coefficients for more robustness. Table 1 shows performance of the proposed embedding, in terms of PSNR (dB) values, in comparison with [2, 12] when all algorithms embed a binary pseudorandom sequence of 128 bits. Dashed lines show PSNR for that image is not reported in the mentioned reference.

We performed another set of experiments for comparing SSIM values of these methods. In [12], only an average SSIM value of 0.9980 is mentioned for ten well-known standard images. We experimented with the same set of images and same watermark length and an average SSIM value of 0.9991 was measured. The SSIM values of our method and method of [2] are summarized in Table 2.

In [17], size of the watermark sequence is 16 bits. To compare PSNR and SSIM values we embed the same size watermark. Results are shown in Table 3. Hence, the results and comparisons for different invisibility tests show that the proposed framework, which asks to embed redundantly in different frequency zones, could successfully preserve perceptual quality of the image for different-size messages. Distortions are negligible due to the error diffusion property of the transform function.

**Fig. 6** **a** Original example images, and **b** watermarked images



Original Lena



PSNR=46.52  
SSIM=0.9996



Original Boat



PSNR=45.67  
SSIM=0.9990



Original Plane



PSNR=44.72  
SSIM=0.9988



Original Peppers



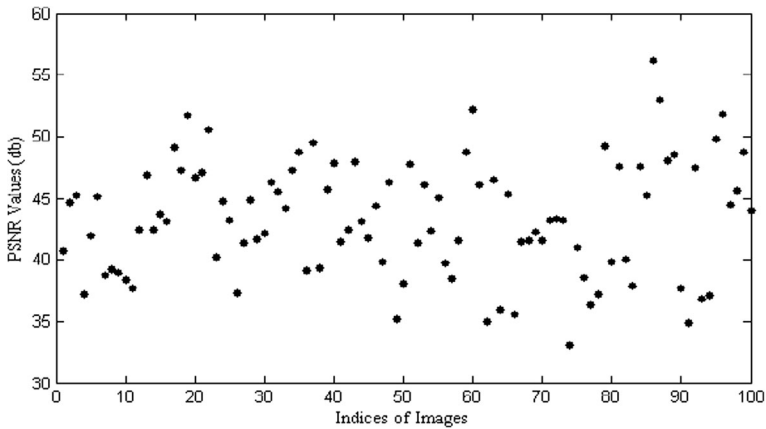
PSNR=48.56  
SSIM=0.9998

(a)

(b)

## 4.2 Watermark robustness

A robust watermarking system should protect the watermark data and should tolerate different attacks that the image may go through. Therefore, we test our framework,



**Fig. 7** PSNR values (dB) for 100 watermarked images by the proposed method (size of the watermark = 128 bit)

using the sample implementation, against some signal processing attacks, different noise attacks, and geometrical attacks. We also compare results with those of other watermarking techniques. In the followings, bit error rate (BER) is used for measuring the robustness of watermark. It represents the percentage of the watermark string of bits that are extracted incorrectly.

At first, we investigate effects of a wide range of attacks on our watermarked images and compare them with methods in [2, 12]. These references have reported results on four images and hence we tested our method on the same images with the same watermark length. The PSNR, SSIM [25], and BER results are shown in Table 4. A 128-bit pseudorandom watermark is embedded in these images. For the additive white Gaussian noise (AWGN) attack, results are for two different noise distributions. Table 4 illustrates that our method has high has the best image quality both in terms of PSNR and SSIM. Our method has about 4 dB higher quality than [2]. Furthermore, our method produces zero BER against Gaussian, crop, and scale attacks, while the compared methods produce bit error rates as big as 4.72. As expected, our method has higher robustness against a wide range of attacks regardless of the part of the frequency spectrum that the attack destroys. Both watermarking methods of [2, 12]

**Table 1** PSNR values of the proposed method and [2, 12] for a 128-bit pseudorandom watermark

Image Name	Hamghalam et al. [12]	Akhaee et al. [2]	Proposed
Lena	43.45		<b>46.52</b>
Goldhill	45.12		<b>47.93</b>
Bridge	<b>43.20</b>	40.85	42.61
Peppers	42.71		<b>48.56</b>
Plane		40.39	<b>44.72</b>
Pirate		40.44	<b>46.19</b>
Boat		39.89	<b>45.67</b>

Best values are bolded

**Table 2** SSIM comparison between the proposed method and [2] for 128-bit pseudorandom watermarks

Image Name	Akhaee et al. [2]	Proposed
Lena		0.9996
Goldhill		0.9996
Bridge	0.9983	<b>0.9985</b>
Peppers		0.9998
Plane	0.9984	<b>0.9988</b>
Pirate	0.9982	<b>0.9993</b>
Boat	0.9984	<b>0.9990</b>

Best values are bolded

insert the message into low frequency coefficients, hence, they do not perform well against high-pass and sharpening filters. We will show that our method is robust against sharpening filter too.

Since the watermarking scheme in [15] inserts data in high frequency DCT coefficients, we compare the performance of this method with that of ours. Watermark in this method is a 64-bits pseudorandom watermark. This comparison is done on USC-SIPI image database [23]. They fix the PSNR value of the watermarked images to 42 dB. In order to make a fair comparison, we provide the same conditions for our experiments. The mean of PSNR values in our method for the images of this database is equal to 45.68 dB which is much better than [15]. Hence, in terms of invisibility we outperform this method. We also compare our method with the method in [11]. This paper uses 2 levels of CT transform for error diffusion and a better transparency. To have a fair comparison between our method and the method in [11], we tested their algorithm on the images of the database. Also, we modified the strength factor of [11] to make their images have similar PSNR to those of our algorithm. Table 5 contains BER results of our method and those of [11, 15]. Watermark for all three methods is a 64-bit pseudorandom watermark. Better results are bolded. The proposed method performs better results under most of the attacks. For some attacks, such as low quality JPEG compression, our method produces lower but comparable results. [15].

To evaluate the performance of our method for very low bit rate embedding, we test robustness of the proposed method for a 16-bit binary watermark string. The results of our method are compared with the method of [17], as shown in Table 6. The PSNR values of [17] on its dataset, which contains 100 watermarked images, are

**Table 3** PSNR and SSIM comparison of the proposed method and [17] for 16-bit pseudorandom watermarks

Image Name	Nasir, et al. [17]		Proposed	
	PSNR	SSIM	PSNR	SSIM
Lena	50.82	0.9996	<b>51.46</b>	<b>0.9998</b>
Peppers	50.87	0.9996	<b>55.10</b>	<b>~1</b>
Baboon	<b>49.46</b>	<b>0.9993</b>	49.21	0.9992

Best values are bolded

**Table 4** PSNR, SSIM and BER values of the proposed method and [2, 12] for a pseudorandom watermark with length 128 bit

Method	SSIM	PSNR (dB)	BER		Rotation (5°)	Gaussian (3 × 3)	Crop (5%)	Scale (0.75)	AWGN (σ = 10)
			JPEG (20%)	(50%)					
Hamghalam et al. [12]	0.9985	43.21	8.01	4.80	4.77	4.72	4.06	0.88	7.97
Akhace et al. [2]	0.9981	40.39	<b>1.92</b>	<b>1.03</b>	<b>2.27</b>	1.18	4.29	0.19	<b>4.27</b>
Proposed	<b>0.9994</b>	<b>44.48</b>	10.53	7.81	11.87	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>7.35</b>

Best values are bolded

**Table 5** BER of the proposed method and scheme in [15] for a 64-bit pseudorandom watermark

Method	JPEG (10%)	Median (3 × 3)	Mean (7 × 7)	Cropping (60%)	Resizing (512–200- 512)	Resizing (512–100- 512)	Rotation (+0.5°)	Rotation (−0.25°)	Sharpen Filter	Salt & Pepper (d = 0.1)
Li et al. [15]	<b>0.00</b>	<b>0.00</b>		8.58	8.38	31.53	22.84	4.09	2.02	
Fazlali et al. [11]	7.23	2.26	6.09	6.74	9.07	32.20	10.94	3.58	8.71	5.17
Proposed	5.34	0.66	<b>2.2</b>	<b>5.12</b>	<b>0.21</b>	<b>17.91</b>	<b>7.55</b>	<b>1.89</b>	<b>0.00</b>	<b>3.01</b>

Best values are bolded



**Table 6** Success rate (%) of the proposed method and scheme in [17] for a 16-bit pseudorandom watermark

Method	JPEG (20%)	Median (3 × 3)	Gaussian (5 × 5)	Wiener (3 × 3)	Centered- Cropping (20%)	CUT, 5rows 17columns	Scaling (0.7)	Rotation (5°)
Nasir et al. [17]	88	95	99	97	94	92	93	95
Proposed	83	<b>97</b>	<b>100</b>	97	<b>95</b>	<b>95</b>	<b>100</b>	86

Best values are bolded

between 41.78 dB and 56.29 dB. The PSNR values in the proposed method, on the same dataset, are between 43.10 dB and 61.21 dB. Reference [17] uses “success rate” to measure robustness, which is defined as the number of images that the watermark is correctly extracted from. For the majority of attacks we perform better, but for some attacks, such as ‘JPEG’ and “rotation” results of [17] are better.

Regarding the computation complexity of the proposed framework it should be noted that complexity is mainly important in real-time applications. In such applications the embedding procedure is of main concern. The complex part of this framework is in the extraction stage and the embedding procedure is relatively straight forward. Procedures, such as feature extraction, use of classification, and voting, are performed in the extraction stage. The extraction of watermark, in most applications, is done offline. The proposed framework’s embedding complexity could be similar to other methods. For example if a DCT based embedding is used, for  $N \times N$  images, the computational complexity would be  $O(N^2 \log_2(N))$  [18].

## 5 Conclusions

In this paper a framework for robust blind image watermarking was proposed. If the watermark was redundantly embedded in transform coefficients of the image then most attacks only disturb a portion of the coefficients. The framework requires that the extraction be performed on less disturbed coefficients. This is done by assessment of watermarked images without knowing the original image. The proposed framework had different stages, such as image transform, embedding, feature extraction, classifier, extraction, and voting. It was shown that the framework was not bounded to, for example, a particular embedding scheme or a particular classifier. This framework required that a classifier be trained to distinguish which part of the frequency spectrum was more disturbed by an attack. It was claimed that such structure would lead to better extraction of data from images that were attacked.

To test our hypothesis, we presented a new embedding scheme. We also proposed a new feature extraction method which revealed how an image was affected by an attack. We trained a classifier using a set of images and attacks. The images that were used for testing of the system were different than those that had been used for the training of the classifier. We showed that the framework, even with a simple embedding method, was robust against a wide range of attacks and its performance was better than some state-of-the-art algorithms. These results could be improved by using more complex and more efficient elements within the proposed framework.

## References

1. Akhaee MA, Sahraeian S, Sankur B, Marvasti F (2009) Robust scaling-based image watermarking using maximum-likelihood decoder with optimum strength factor. *IEEE Transactions on Multimedia* 11(5):822–833
2. Akhaee MA, Sahraeian SME, Craig J (2011) Blind image watermarking using a sample projection approach. *IEEE Transactions on Information Forensics and Security* 6(3):883–893
3. Al-Haj A (2007) Combined DWT-DCT digital image watermarking. *J Comput Sci* 3(9):740–746
4. Azzari L, Foi A (2014) Indirect estimation of signal-dependent noise with non-adaptive heterogeneous samples. *IEEE Trans Image Process* 23(8):3459–3467
5. Burgett S, Koch E, Zhao J (1994) A novel method for copyright labeling digitized image data. Technical Report of Fraunhofer Institute for Computer Graphics, Darmstadt
6. Chen YH, Huang HC (2015) Coevolutionary genetic watermarking for owner identification. *Neural Comput & Applic* 26(2):291–298
7. Chu SC, Huang HC, Shi Y, Wu SY, Shieh CS (2008) Genetic watermarking for zerotree-based applications. *Circuits, Systems & Signal Processing* 27(2):171–182
8. Corel Draw Software and Dataset available at: <http://www.corel.com>
9. Craver S, Memon N et al (1998) Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications. *IEEE Journal on Selected Areas in Communications* 16(4):573–586
10. Fan Y, Zhu YS, Liu Z (2016) An improved SIFT-based copy-move forgery detection method using T-linkage and multi-scale analysis. *Journal of Information Hiding and Multimedia Signal Processing* 7(2):399–408
11. Fazlali HR, Samavi S, Karimi N, Shirani S (2016) Adaptive blind image watermarking using edge pixel concentration. *Multimedia Tools and Applications* 1:16
12. Hamghalam M, Mirzakuchaki S, Akhaee MA (2014) Geometric modelling of the wavelet coefficients for image watermarking using optimum detector. *IET Image Process* 8(3):162–172
13. Jiang J, Feng G (2002) The spatial relationship of DCT coefficients between a block and its sub-blocks. *IEEE Trans Signal Process* 50(5):1160–1169
14. Kang X, Huang J, Zeng W (2008) Improving robustness of quantization-based image watermarking via adaptive receiver. *IEEE Transactions on Multimedia* 10(6):953–959
15. Li Z, Yap KH, Lei BY (2011) A new blind robust image watermarking scheme in SVD-DCT composite domain. *Proc. of IEEE International Conference on Image Processing (ICIP)* 2757–2760
16. Miller ML, Cox IJ, Linnartz JP, Kalker T (1999) A review of watermarking principles and practices. *Digital Signal Processing for Multimedia Systems*:461–485
17. Nasir I, Khelifi F, Jiang J, Ipson S (2012) Robust image watermarking via geometrically invariant feature points and image normalisation. *IET Image Process* 6(4):354–363
18. Oppenheim AV, Schaffer RW (2010) *Discrete-time signal processing*. Pearson Higher Education
19. Podilchuk CI, Delp EJ (2001) Digital watermarking: algorithms and applications. *IEEE Magazine, Signal Processing* 18:33–46
20. Subramanyam AV, Emmanuel S, Kankanhalli MS (2012) Robust watermarking of compressed and encrypted JPEG2000 images. *IEEE Transactions on Multimedia* 14(3):703–716
21. Surekha B, Swamy GN (2011) A spatial domain public image watermarking. *International Journal of Security and Its Applications* 5(1):1–12
22. Tabatabaei SA, Ur-Rehman O, Zivic N (2015) Secure and robust two-phase image authentication. *IEEE Transactions, Multimedia* 17(7):945–956
23. The USC-SIPI Image Database Volume 3, available at: <http://sipi.usc.edu/database/database.php?volume=misc>
24. Wang Y, Moulin P (2007) Optimized feature extraction for learning-based image steganalysis. *IEEE Transactions on Information Forensics and Security* 2(1):31–45
25. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612
26. Wang S, Miyauchi R, Unoki M, Kim NS (2015) Tampering detection scheme for speech signals using formant enhancement based watermarking. *Journal of Information Hiding and Multimedia Signal Processing* 6:1264–1283
27. Xinshan Z, Ding J, Dong H et al (2014) Normalized correlation-based quantization modulation for robust watermarking. *IEEE Transactions on Multimedia* 16(7):1888–1904



**Morteza Heidari** received his Bachelor of Science degree in Electrical Engineering, majoring in Communication, from Isfahan University of Technology in 2011. He also received his Master of Science degree in the same field from Sharif University of Technology in 2013. His research interests are machine learning, image and signal processing, and watermarking.



**Shadrokh Samavi** is a Professor of Computer Engineering at Isfahan University of Technology, Iran. He is also an Adjunct Professor at the ECE department of McMaster University where he is a member of the Multimedia Signal Processing Lab. Dr. Samavi is also a research affiliate at the Biomedical and Clinical Informatics Lab, Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, Michigan, USA. Professor Samavi completed a B.S. degree in Industrial Technology and received a B.S. degree in Electrical Engineering at California State University, a M.S. degree in Computer Engineering at the University of Memphis and a Ph.D. degree in Electrical Engineering at Mississippi State University, U.S.A. Dr. Samavi is a Registered Professional Engineer (PE), USA. He is also a member of IEEE and a member of Eta Kappa Nu and Tau Beta Pi honor societies. Shadrokh Samavi's research interests are in the areas of image processing and hardware implementation and optimization of image processing algorithms. He is also interested in compression and processing of biomedical images, as well as, VLSI design and computer arithmetic.



**S.M. Reza Soroushmehr** is a postdoctoral fellow at University of Michigan, Ann Arbor, MI, USA. Prior to this position he was a postdoctoral fellow at Electrical and Computer Engineering (ECE) department of McMaster University, Hamilton, ON, Canada. He received his B. Sc, M. Sc. and Ph.D. (with honor) respectively in 2000, 2004 and 2013 from the ECE department of Isfahan University of Technology (IUT), Isfahan, Iran. His main research interests include image processing, video compression, algorithm design and optimization.



**Shahram Shirani** (SM'04) received the B. Sc. degree in electrical engineering from the Isfahan University of Technology, Isfahan, Iran, in 1989, the M. Sc. degree in biomedical engineering from the Amirkabir University of Technology, Tehran, in 1994, and the Ph.D. degree in electrical and computer engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. Since July 2000, he has been with the department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada, where he is currently a full Professor. His research interests include image and video compression, multimedia communications, and ultrasonic imaging. Dr. Shirani is a registered Professional Engineer (P. Eng.).



**Nader Karimi** received the B.S. degree (summa cum laude) in Computer Engineering from Azad University, Arak, Iran, in 2002. He received his M. Sc. (honor) in Computer Engineering, and Ph.D. degree (honor) in Electrical Engineering from Isfahan University of Technology (IUT), Iran, in 2004 and 2012, respectively. He is currently an Assistant Professor at the Department of Electrical and Computer Engineering, Isfahan University of Technology. His research interests are image compression, hardware implementation and optimization of image processing algorithms, and watermarking.



**Kayvan Najarian** received the B. Sc. degree in electrical engineering from Sharif University, Tehran, Iran, the M. Sc. degree in biomedical engineering from Amirkabir University, Tehran, and the Ph.D. degree in electrical and computer engineering from the University of British Columbia, Vancouver, Canada. He is an Associate Professor in the Departments of Computational Medicine and Bioinformatics, and Emergency Medicine at the University of Michigan, Ann Arbor, MI, USA. He also serves as the Director of the Michigan Center for Integrative Research in Critical Care's Biosignal- Image and Computational Core program. His research interests include design of signal/image processing and machine learning methods to create computer assisted clinical decision support systems that improve patient care. Dr. Najarian serves as the Editor-in-Chief of a journal in the field of biomedical engineering as well as the Associate Editor of two journals in the field of biomedical informatics. He is also a Member of many editorial boards and has served as a Guest Editor of special issues for several journals.