BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 60, No. 2, 2012
DOI: 10.2478/v10175-012-0042-5

On domain selection for additive, blind image watermarking

P. LIPIŃSKI*

Institute of Information Technology, Technical University of Lodz, 215 Wólczańska St., 90-924 Łódź, Poland

Abstract. Recently, a variety of digital watermarking schemes have been developed for copyright protection of digital images. In robust watermarking, which is used in copyright protection, transform-based algorithms are used to ensure resilience of the watermark to common signal processing attacks. The most frequently used watermarking algorithms for additive watermark embedding involve DCT, DFT, SVD and DWT domains. In this article we verify which domain is optimal for robust, the additive watermark embedding scheme. We demonstrate that in additive watermark embedding the embedding domain plays more important role than the embedding formula.

Key words: watermarking, digital watermarking, robust watermarking.

1. Introduction

Nowadays, most information is stored and processed in digital form on a computer. Unfortunaltely, digital content can be easily copied and distributed which makes illegal copying and disrtibution of digital content extremely easy. Therefore, there is a strong need for developing techniques which can help to identify intellectual property of digital documents [1].

Digital Watermarking is one of possible technologies which can be used for protecting intellectual property of digital content [2–5]. The research performed by the author, published in [6] has demonstrated that there is no commercial software nor technology which satisfies all requirements of robust digital watermarking. As a result there is a strong need to develop new robust watermarking algorithms.

Robustness of watermarking schemes is usually ensured by embedding the watermark in transform domain. The transform should be selected in such a way that it is attack invariant, that is, the attack should not change the watermark. In practical applications Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD) are used. Most authors focus on a single watermark embedding scheme in a single domain and compare the scheme with a traditional one, usually in the same domain [3,4,7]. As a result, it is very difficult to determine which watermark embedding scheme performs actually best in terms of robustness and how much the embedding domain influences the robustness of the scheme. Therefore, here we verify which domain is optimal for watermark embedding.

The article is organized as follows. In Sec. 2 we briefly describe digital watermarking in transform domain, in Sec. 3 we define transforms which are used in transform-based watermark embedding, in Sec. 4 we demonstrate experimental results and explain the testing procedure, in Sec. 5 we give conclusions.

2. Digital watermarking scheme in transform domain

For the purpose of this article we make a clear distinction between watermark embedding scheme (the whole algorithm) and the watermark embedding formula (the formula which is used for adding the watermark in a domain) – see Fig. 1.

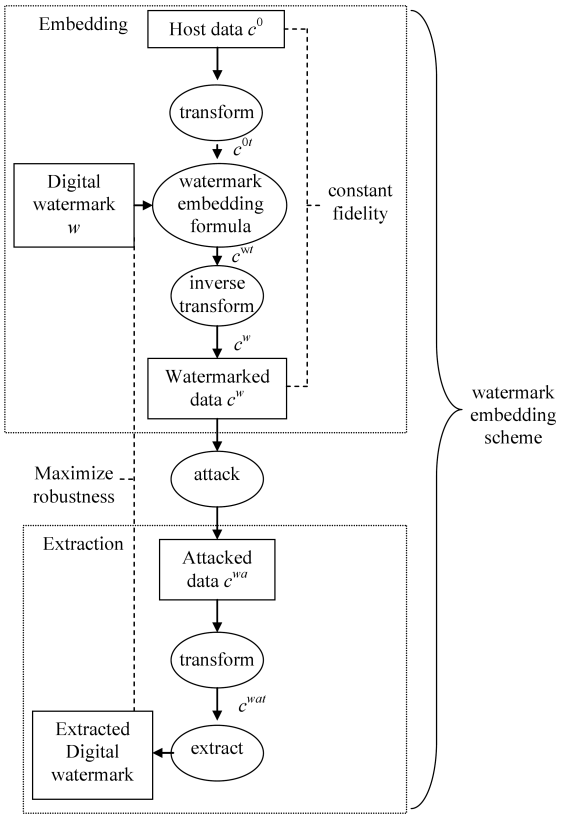

Fig. 1. Scheme of transform-based digital watermarking

The general transform-based watermark embedding scheme is given as follows: first, *host data* c_0 is transformed

*e-mail: piotr.kazimierz.lipinski.gmail.com

into chosen domain (DCT, DFT, DWT, SVD), next a *digital watermark* w is embedded into most prominent coefficients of the *transformed host data* c^{0t} using one of the following formulas:

$$c^{wt} = c^{0t} + \alpha \left| c^{0t} \right| \cdot w, \tag{1}$$

$$c^{wt} = c^{0t} + \alpha \cdot c^{0t} \cdot w, \tag{2}$$

$$c^{wt} = c^{0t} + \alpha \cdot w, \tag{3}$$

$$c^{wt} = c^{0t} \cdot e^{\alpha w}, \tag{4}$$

where α – embedding strength, c^{wt} – watermarked data in transform domain.

Each equation (1)–(4) constitutes different watermark embedding formula.

Here we consider multibit watermarking, where the watermark w is a K-element random series: $w \in [-1, 1]$. Last, an *inverse transform* of the watermarked data is performed. As a result one obtains a *watermarked data* c^{wt} (Fig. 1). After the embedding process the watermarked data is usually attacked (otherwise there is no point in carrying out experiments) [2]. In this article we focus on image processing attack such as: white noise, resize, median filtering, low-pass filtering, Gaussian noise, jpeg compression, color depth reduction which are commonly used by photographers and web developers, see [6].

To detect the presence of the watermark, *attacked data* c^{wa} is transformed into a chosen domain and *extraction algorithm* is applied to watermarked domain-based data (Fig. 1). The extraction algorithm uses normalized correlation C between the *watermark* w and most prominent coefficients of attacked data in transform domain c^{wat} [1]:

$$C = \frac{1}{K-1} \sum_{i=1}^{K} \frac{\left(c_i^{wat} - \overline{c}_i^{wat}\right)(w_i - \overline{w})}{\sigma_c \sigma_w}, \tag{5}$$

where σ_c – standard deviation of watermarked coefficients, σ_w – standard deviation of the watermark, c_i^{wat} – i-th most prominent coefficient of the attacked data, $\overline{c}^{wat}$ – mean value of the most prominent coefficients of the attacked data, w_i – i-th element of the embedded watermark, $\overline{w}$ – mean value of the embedded watermark, K – watermark length.

When C is above threshold we assume that the watermark is detected.

When investigating performance of watermarking schemes three conflicting performance metrics of watermarking system must be taken into consideration: fidelity, robustness and payload size. For copyright protection payload size can be regarded as fixed parameter [5], that is, in all experiments we embed the watermark of the same length. In watermark embedding we take advantage of iterative algorithm which ensures constant fidelity for all images and all embedding domains. This allows us to measure separability s as a robustness measure, defined as:

$$s = \min\left(C - C_k\right), \tag{6}$$

where $k = 1, 2, \ldots, M$, C – correlation between the embedded watermark and the most prominent coefficients of attacked data in transform domain, C_k– correlation between the k-th random watermark and the most prominent coefficients of attacked data in transform domain.

Separability can be interpreted as difference between correlation of the watermark and the highest correlation with randomly generated watermark (see Fig. 2). The higher the separabilty the higher the robustness of the embedding scheme.

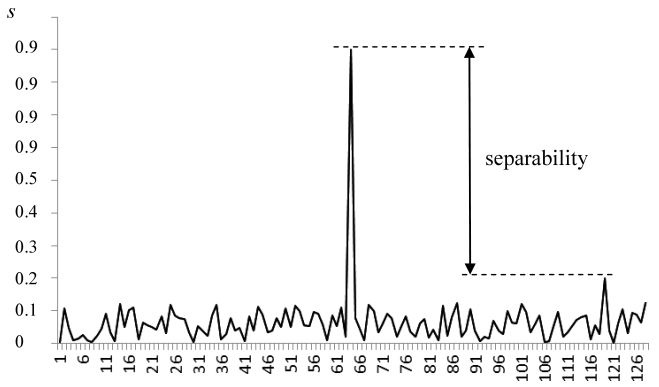

Fig. 2. Separability – the measure of robustness of a watermarking algorithm

The main purpose for using transform-based embedding is the possibility of selecting only those samples of the transform-domain watermarked which have desired properties in terms of fidelity and robustness. For example in DCT or DFT-based watermarking, samples corresponding to middle frequencies are used to embed watermark usually, in DWT-based watermarking only high frequency components from levels 2–4 of wavelet transform are used for watermark embedding [3, 4]. This is mainly due to the fact that watermarks cannot be embedded in low frequency components, as this would deteriorate the fidelity, on the other hand high frequency components cannot be used as they are susceptible to attacks.

3. Transforms

The watermark embedding scheme from Sec. 2 is general and can take advantage of any transforms. Here we define those transformations which are frequently used in watermark embedding, that is: DCT, DFT, DWT, SVD and give transformation-specific information. In order to verify which domain gives most promising results we use four different watermarking formulas (1)–(4) for watermark embedding and the same formula (5) for watermark extraction in all cases.

3.1. DCT algorithm. The first algorithm for transformed-based watermark embedding was introduced by Cox et al. in [2]. In the first version of the algorithm, the $M \times N$ image (*host data*) is transformed into DCT domain using the formula:

$$c_{DCT}^{0t}[k,l] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} c_0[m,n] \cdot \cos\left(\frac{-\pi m k}{M}\right) \cdot \cos\left(\frac{-\pi n l}{N}\right), \tag{7}$$

where $c^{0t}_{DCT}[k,l]$ is a DCT transform of the *host data* $c_0[m,n]$.

Next, embedding formulas (1)–(4) from Sec. 2 are used to add a watermark in DFT domain. Cox, in his original paper, used only the formula (1) for watermark embedding, but to make the comparison possible we use all formulas (1)–(4). After embedding, c^{wt} is transformed to spatial domain using formula (8) and we obtain watermarked image c^w:

$$c^w_{DCT}[k,l] = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} c^{wt}_{DCT}[k,l] \cdot \cos\left(\frac{\pi m k}{M}\right)\cdot\cos\left(\frac{\pi n l}{N}\right), \tag{8}$$

where $c^w_{DCT}[k,l]$ – watermark data (using DCT algorithm), $c^{wt}_{DCT}[k,l]$ – watermarked data in DCT domain.

In the detection algorithm the attacked, watermarked data $c^{wa}_{DCT}[k,l]$ is transformed to DCT domain using the formula (7). The correlation between the watermark w and K most prominent DCT coefficients is calculated using the formula (5). We assume that the watermark is present, when the correlation coefficient is above the threshold. In our experiments we use the formula (6) to compute separability.

3.2. DFT algorithm. In the same article [2] Cox et al. consider using DFT transform. The algorithm is analogical to DCT based, but DCT transform is replaced with DFT transform given by:

$$c^{0t}_{DFT}[k,l] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} c_0[m,n]\cdot e^{\frac{-j2\pi mk}{M}}\cdot e^{\frac{-j2\pi nl}{N}} \tag{9}$$

and IDFT transform:

$$c^w_{DFT}[k,l] = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} c^{0t}_{DFT}[m,n] \cdot e^{\frac{-j2\pi mk}{M}}\cdot e^{\frac{-j2\pi nl}{N}}, \tag{10}$$

where $c^w_{DFT}[k,l]$ – watermark data (using DFT algorithm), $c^{wt}_{DFT}[k,l]$ – watermarked data in DFT domain.

The watermark is added to those coefficients in DFT domain which have the most prominent magnitude. The watermarking algorithm must take into consideration that the image to be real imposes, the following constraint on the values of DFT:

$$c^{0t}_{DFT}[k,l] = c^{0t^*}_{DFT}[M-k,N-l]. \tag{11}$$

Therefore, when watermarking, changes of magnitude must preserve positive symmetry:

$$\begin{aligned} c^w_{DFT}[k,l] &= c^{0t}_{DFT}[k,l] + \alpha\cdot w\cdot c^{0t}_{DFT}[k,l], \\ c^w_{DFT}[M-k,N-l] &= c^{0t}_{DFT}[M-k,N-l] \\ &\quad +\alpha\cdot w\cdot c^{0t}_{DFT}[M-k,N-l]. \end{aligned} \tag{12}$$

Ramkuar et al., noticed in [8] that the most information about an image is contained in the phase of DFT. Adding the watermark to the most prominent coefficients of phase of the DFT was proposed by Ruanaidh in [9]. The watermarking algorithm must take into consideration that the image to be real imposes the condition (11). So when watermarking, changes of phase must preserve negative symmetry:

$$\begin{aligned} c^w_{DFT}[k,l] &= c^{0t}_{DFT}[k,l] + \alpha\cdot w\cdot c^{0t}_{DFT}[k,l], \\ c^w_{DFT}[M-k,N-l] &= c^{0t}_{DFT}[M-k,N-l] \\ &\quad +\alpha\cdot w\cdot c^{0t}_{DFT}[M-k,N-l]. \end{aligned} \tag{13}$$

3.3. DWT algorithm. In Discrete Wavelet Transform a signal is split into two parts of high and low frequencies. High frequencies correspond to the edge components of the signal. Low frequencies are again split into two parts of low and high frequencies. This process can be continued arbitrary number of times. From these DWT coefficients, the original signal can be reconstructed. The process of reconstruction is called Inverse Discrete Wavelet Transform (IDWT). In two-dimensional case (images) DWT is defined by implementing one dimensional DWT for each dimension. The 3-level pyramid structure of two-dimensional DWT is shown in Fig. 2. The watermarking algorithm taking advantage of DWT was introduced by Xia et al. in [10]. In his original algorithm the watermark was embedded into most prominent high frequency components of DWT. Here we use formulas (1)–4) to embed the random watermark into K high frequency components which have the largest magnitude on a third level of DWT decomposition and which are not located at the lowest frequency band, (see Fig. 3).

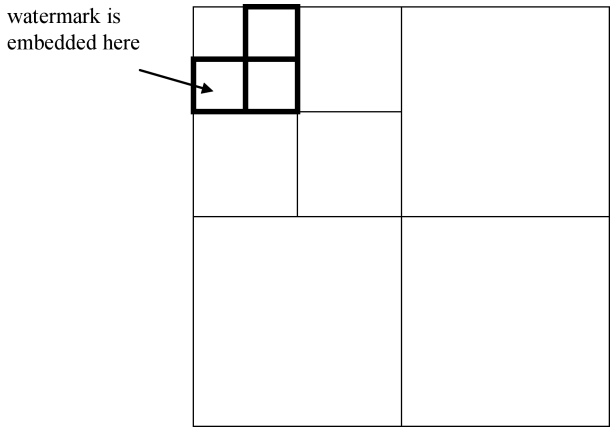

Fig. 3. DWT-based watermark embedding

In the detection algorithm the attacked watermarked data $c^{wa}_{DWT}[k,l]$ is transformed to 3-level DWT. The correlation between the watermark w and K coefficients which have the largest magnitude on a third level of DWT decomposition is calculated using the formula (5). We assume that the watermark is present, when the correlation coefficient is above threshold. In our experiments we use the formula (6) to compute separability.

3.4. SVD algorithm. Discrete image can be treated as an array of nonnegative scalar entries, which can be regarded as

a matrix $\mathbf{C}_0 \in \mathbf{R}^{N\times N}$. Singular Value Decomposition of $\mathbf{C}_0$ is defined as:

$$\mathbf{C}_0 = \mathbf{U}\mathbf{C}^{0t}_{SVD}\mathbf{V}^*, \tag{14}$$

where $\mathbf{U} \in$R – unitary matrix, $\mathbf{V} \in$R – unitary matrix, $()^*$ – conjugate transpose, $\mathbf{C}^{0t}_{SVD}$ – diagonal matrix.

There are several watermarking algorithms using the formula (14). Here we use blind version of the embedding algorithm presented in [1]. We add the watermark $\mathbf{W}$ into diagonal matrix $\mathbf{C}^{0t}_{SVD}$ using the formulas (1)–(4). Watermarked image is calculated from:

$$\mathbf{C}_w = \mathbf{U}\mathbf{C}^{wt}_{SVD}\mathbf{V}^*, \tag{15}$$

where $\mathbf{C}^{wt}_{SVD}$ – watermarked diagonal matrix, C_w – watermarked image.

To extract the watermark singular value decomposition of attacked watermarked image C_{wa} is calculated:

$$\mathbf{C}_{wa} = \mathbf{U}\mathbf{C}^{wat}_{SVD}\mathbf{V}^*. \tag{16}$$

The correlation between the watermark $w$and diagonal values of $\mathbf{C}^{wt}_{SVD}$ is calculated using the formula (5). We assume that the watermark is present, when the correlation coefficient is above threshold. In our experiments we use the formula (6) to compute separability.

4. Experimental results

In this section, we use the watermark embedding scheme from Fig. 1 to embed the watermark in five different domains. The aim of these experiments is to choose the best domain in terms of robustness at constant fidelity and capacity for additive watermark embedding.

In our experiments we use 64 images from USC-SIPI Image Database. Each image is a gray scale, 512×512 pixels size, 8 bit gray. Pictures are selected from textures, aerials, miscellaneus and sequences. Sample pictures from the database are shown in Fig. 4. They reflect various picture characteristics such as frequency, sharpness etc. Watermarks are generated as 512 element vectors where $w = [-1, 1]$. Separability is computed using a set of 1000 randomly generated 512-element vectors. Each test is repeated using 5 different watermarks and 5 different randomly generated watermark sets.

We compare the separability, as a measure of the watermark robustness against signal processing attacks, for different watermark embedding domains. The experiments are carried out in the following way: First, each picture is transformed to a chosen domain: DCT, DWT, SVD, DFT. Second, the watermark is embedded using the formulas (1)–(4). Third, the inverse transform of the watermark is computed. In DCT transform two cases are considered: the watermark was embedded in magnitude or phase of the host data. In DWT, Daubechies 6 wavelet transform is used and the watermark is embedded in third level of decomposition. We use adaptive algorithm to adjust the embedding strength α so as to ensure constant fidelity for each domain. PSNR is used a as a fidelity measure. In other words, each watermarked image has the same PSNR value. Fourth, we attack the image taking advantage of the following: white noise, resize, median filtering, low-pass filtering, Gaussian noise, jpeg compression, color depth reduction. In order to extract the watermark each attacked image is transformed by the means of corresponding transform. Last, the separability coefficient is computed for each transformed image from (6). Separability coefficients for Lena image at fidelity 35dB are presented in Table 1. The results for other test images are very close to the results for Lena image. Due to lack of space the results for all test pictures cannot be presented here.

It can be easily noticed that DWT perform best in terms of separability at constant fidelity level for all signal processing attacks presented here. Furthermore, DWT outperforms other domains for all attacks and for all embedding formulas. This is mainly due to the fact that it allows optimal space and frequency localization of a watermark, which means that the watermark can be optimally located in space and frequency simultaneously. Separability is clearly best in spatial-domain embedding when the image is not attacked, but it performs much worse after attacks. This leads to the general conclusion that DWT is optimal for additive watermark embedding when the image is attacked using signal processing attacks.

Fig. 4. Sample test pictures

Table 1
Separability coefficients for Lena image at fidelity 35 dB

	No attack	Color depth reduction	Jpeg compression	Gaussian noise	Low-pass filtering	Median filtering	Resize 50%	White noise
				Formula (1)				
Spatial domain	**0.4235**	0.1022	0.1250	0.1530	0.0897	0.1082	0.1528	0.0893
DWT	0.3861	**0.3765**	**0.3340**	**0.2644**	**0.3314**	**0.3188**	**0.3308**	**0.3740**
DFT phase	−0.2930	−0.2581	−0.2752	−0.2938	-0.3627	-0.2307	−0.2132	−0.3406
DFT magnitude	0.1391	0.1389	0.1389	0.1342	0.1325	0.1365	0.1384	0.1394
DCT	0.0896	0.0899	0.0897	0.0875	0.0895	0.0881	0.0862	0.0884
SVD	0.0131	0.0141	−0.2195	0.0046	0.0114	0.0115	−0.1586	0.0052
				Formula (2)				
Spatial domain	0.4056	0.2165	0.1936	0.1702	0.2257	0.2183	0.2237	0.2577
DWT	**0.4385**	**0.3991**	**0.3195**	**0.3205**	**0.3763**	**0.3765**	**0.3038**	**0.4138**
DFT phase	−0.1751	−0.2207	−0.2025	−0.2361	−0.2424	−0.1674	−0.1856	−0.2939
DFT magnitude	0.1866	0.1419	0.1482	0.2129	0.1222	0.2218	0.1710	0.2122
DCT	0.1338	0.1478	0.1122	0.0973	0.1626	0.1714	0.0793	0.1299
SVD	0.0434	0.0482	−0.1364	0.0553	0.0859	0.0678	−0.1148	0.0917
				Formula (3)				
Spatial domain	**0.3528**	0.2076	0.1603	0.1760	0.1435	0.1512	0.2039	0.1624
DWT	0.3387	**0.2934**	**0.2848**	**0.2080**	**0.2695**	**0.2835**	**0.2781**	**0.3784**
DFT phase	−0.2034	−0.0977	−0.1273	−0.1543	−0.2713	−0.1387	−0.1489	−0.2321
DFT magnitude	0.1163	0.1871	0.1419	0.1074	0.1297	0.1026	0.1161	0.1087
DCT	0.1274	0.0902	0.1599	0.0867	0.0866	0.1500	0.1274	0.0978
SVD	0.0897	0.0270	−0.0831	0.0295	0.0355	0.0987	−0.1188	0.0893
				Formula (4)				
Spatial domain	**0.2943**	0.2164	0.1689	0.1591	0.1707	0.1974	0.1808	0.1654
DWT	0.2672	**0.3004**	**0.2623**	**0.2340**	**0.1896**	**0.2718**	**0.2464**	**0.2960**
DFT phase	−0.0649	−0.0704	−0.1169	−0.1212	−0.1811	−0.0487	−0.0610	−0.1045
DFT magnitude	0.0953	0.1550	0.0920	0.1005	0.1565	0.1515	0.1552	0.0937
DCT	0.0992	0.1188	0.0937	0.1422	0.1106	0.1406	0.0989	0.1030
SVD	0.0991	0.0932	−0.1096	0.0767	0.0466	0.0281	−0.0629	0.0906

5. Conclusions

The experimental verification which of the following domains: *DCT, DFT, SVD* and *DWT* is optimal for additive, blind watermark embedding at constant fidelity, has been our aim. For this purpose we have embedded watermarks into 64 images using four different embedding formulas in four abovementioned domains, and tested separability of extracted watermark. The experiments were performed for seven different signal processing attacks. It has been shown that DWT performed best in all experiments. Further research should include adaptive DWT-based algorithms [11], complex domains, such as DWT-SVD and adaptive transforms, with particular emphasis on DWT adaptive transforms.

REFERENCES

[1] D.V.S. Chandra, "Digital image watermarking using singular value decomposition", *Proc. MWSCAS-2002 Circuits and Systems, 45th Midwest Symp.* 3, III-264-III-267 (2002).

[2] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia", *IEEE Trans. on Image Processing* 6 (12), 1673–1687 (1997).

[3] L. Hu and F. Wan, "Analysis on wavelet coefficient for image watermarking", *Proc. Int. Multimedia Information Networking and Security (MINES) Conf.* 1, 630–634 (2010).

[4] H. Yuning, W. Bo, and W. Gang, "A color image fragile watermarking algorithm based on dwt-dct", *Proc. Chinese Control and Decision Conf. (CCDC)* 1, 2840–2845 (2010).

[5] W. Ja-Ling Huang Chun-Hsiang, "Fidelity-guaranteed robustness enhancement of blind-detection watermarking schemes", *Information Sciences* 179, 791–808 (2009).

[6] P. Lipinski. "Watermarking software in practical applications", *Bull. Pol. Ac.: Tech.* 59, 21–25 (2011).

[7] M. Makhloghi, F. Akhlaghian, and H. Danyali, "Robust digital image watermarking using singular value decomposition", *Proc. IEEE Int. Signal Processing and Information Technology (ISSPIT) Symp.* 1, 219–224 (2010).

[8] M. Ramukumar, "A robust data hiding scheme for images using DFT", *Image Processing, ICIP 99. Proc. Int. Conf.* 2, 211–215 (1999).

[9] J.J.K.O. Ruanaidh, W.J. Dowling, and F.M. Boland, "Phase watermarking of digital images", *Proc. Conf. Int. Image Processing* 3, 239–242 (1996).

[10] X. Xiang-Gen, C.G. Boncelet, and G.R. Arce, "A multiresolution watermark for digital images", *Proc. Conf. Int. Image Processing* 1, 548–551 (1997).

[11] J. Stolarek, "Adaptive synthesis of a wavelet transform using fast neural network", *Bull. Pol. Ac.: Tech.* 59, 9–13 (2011).