Noname manuscript No.
(will be inserted by the editor)

An Algorithm for Robust Image Watermarking based on the DCT and Zernike Moments

Budimir Lutovac · **Miloš Daković** ·
Srdjan Stanković · **Irena Orović**

Received: date / Accepted: date

Abstract An image watermarking scheme in the 2D DCT domain is proposed by exploring the advantages of using Zernike moments. Zernike transform has been used in image processing applications such as image recognition, authentication, protection, etc. Here, we propose to use the Zernike moments of the DCT transform to provide an efficient watermarking method. Particularly, the novelty of the proposed approach relies on the method for selection of features that will enable both preserving the image quality and robustness to attacks. Also, a criterion for selection of image blocks suitable for watermarking is given. It is based on the ℓ_1-norm of Zernike moments. The efficiency of the proposed watermarking algorithm is proved on several examples considering different types of attacks (compression, noise, filtering, geometrical attacks).

Keywords Image Watermarking · Region clasification · Zernike moments · 2D DCT transform · Image quality

1 Introduction

Digital watermarking has been widely used to protect the digital multimedia content through the several aspects: authorship protection, copyright protection, authentication, integrity protection, etc. Generally, watermarking is implemented by inserting a certain secret sequence called watermark (random sequences, logos, images) to a specific region of the digital content. This sequence provides information that proves ownership, data integrity and authenticity, or allows tracking of the copies [1–3]. Depending on the particular purpose, the digital watermark can be embedded in a visible or invisible manner, in the sense that it may or may not influence the original data quality. Most of the schemes tend toward imperceptible watermark embedding

Budimir Lutovac(✉), Miloš Daković, Srdjan Stanković, Irena Orović
University of Montenegro, Faculty of Electrical Engineering
Dz. Vasingtona bb, Podgorica, 81000, Montenegro
E-mail: budo@ac.me, milos@ac.me, srdjan@ac.me, irenao@ac.me

that will be at the same time robust to common post-processing techniques called attacks. The attacks may seriously influence the watermark detection and consequently the watermarking purpose.

Watermarking techniques can also be used for data hiding. It is the process used to hide data into an image. Some data hiding schemes such as reversible data hiding are recently proposed in [4,5]. The method with flexible self-recovery quality using compressive sensing concepts to recover the lost content had been proposed in [6]. Data-hiding and compression scheme that combines side-match vector quantization (SMVQ) and image inpainting to obtain both low compression rate and good embedding capacity are proposed in [7]. Although the purpose of data hiding techniques is different from that of traditional watermarking techniques similar methods for data embedding are used in both cases.

The watermark embedding has been implemented using spatial domain or different transformation domains such as DCT domain [8–10], DWT domain [11–13], time-frequency or space/spatial-frequency transformation [14,15]. When considering invisible and robust watermarking scenario, it has been shown that the transform domain based approaches highly overcome the performance of time or spatial domain techniques. Namely, most of the watermarking principles result from the compression principles. For instance, the algorithms for image processing and compression are mostly based on the DCT or DWT domain due to good energy compaction on a small number of coefficients. Moreover, it is known that the DCT coefficients from the low-frequency range contain most of the image energy and certain change made in this range may significantly influence the quality. In contrary, the high-frequency coefficients bring the information about the details and have less influence to the signal quality, but the high-frequency coefficients could be easily removed or degraded by the attacks (e.g., compression). Therefore, most of the known watermarking methods employ the middle frequency regions.

Apart from a suitable choice of watermark embedding domain, it is usually desirable to explore the set of additional image features which might improve the performance of the watermarking procedure in the presence of attacks. For instance, the analysis of image moments is often used in image processing and pattern recognition [16,17]. Image moments are numerical descriptors that contain information about the image properties invariant to certain operations on the image such as translation, rotation, and scaling. In this paper, we consider the use of image moments obtained by mapping the DCT blocks to a set of Zernike polynomials. A detailed description of Zernike moments calculation can be found in [18–20].

The existing watermarking techniques addressing the use of Zernike moments are implemented in the spatial domain [21–24]. In [21–23] the authors use Zernike moments approximation of the whole image for watermark embedding. The problem of watermark embedding directly into the Zernike moments is discussed, including high computational cost and low calculation accuracy of the higher order moments. Furthermore, in [24] the local Zernike moments are applied to the identified circular regions and then the watermark is embedded in each region. Each of the mentioned watermarking schemes is related directly to the spatial domain which is sensitive to attacks and also highly sensitive to the level of change introduced by the watermark.

Here, we propose the transform domain image watermarking procedure using the 2D DCT and Zernike moments. The Zernike moments applied to the transform domain coefficients result in an efficient set of feature used to select the most appropriate regions for watermark embedding. Hence, the proposed method provides automated procedure for suitable regions/coefficients selection. In order to calculate the Zernike moments, the input data have to be located inside the unit circle. Therefore, the proposed algorithm uses four 2D DCT transformations and combines them in a way suitable for analysis using a finite number of Zernike moments. The resulting approximation is used to find the region in the transformation domain, where inserted watermark will not deteriorate the image quality, and will be resistant to attacks. In order to reduce the computational complexity and robustness of watermark for large images, the watermarking procedure is implemented on a block by block basis. Using the ℓ_1-norm of Zernike moments, we also propose a criterion for the selection of blocks suitable for watermarking. The performance of the proposed watermarking scheme is tested and compared with the standard and widely used DCT watermarking procedure. The robustness to several types of attacks is examined including some geometrical attacks such as rotation and trimming of certain image parts.

The paper is organized as follows. The theoretical background on the Zernike moments is given in Section 2. The proposed procedure for the watermarking regions selection and classification is presented in Section 3. The criterion for block selection is given in Section 4. Experimental results and comparisons are presented in Section 5.

2 Theoretical background - Zernike moments

Zernike moments consist of a set of complex polynomials which form a complete orthogonal set over the interior of the unit circle $x^2 + y^2 \leq 1$, [16]. In polar coordinates, these polynomials have the form:

$$V_{nm}(x,y) = V_{nm}(r,\theta) = R_{nm}(r)e^{im\theta} \tag{1}$$

where n is a positive integer; m is an integer that takes positive, negative or zero values such that $n - |m|$ is even and nonnegative. The length of the vector from the origin to the pixel (x,y) is denoted by r. The angle between the vector r and x axis is θ, while $R_{nm}(r)$ are radial polynomials defined as:

$$R_{nm}(r) = \sum_{k=0}^{(n-|m|)/2} \frac{(-1)^k (n-k)!}{k!\left(\frac{n+|m|}{2} - k\right)!\left(\frac{n-|m|}{2} - k\right)!} r^{n-2k} \tag{2}$$

For a function $f(x,y)$ the Zernike moment of order n with repetition m is defined as:

$$A_{nm} = \frac{n+1}{n} \iint_{x^2+y^2 \leq 1} f(x,y) V_{nm}^*(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{3}$$

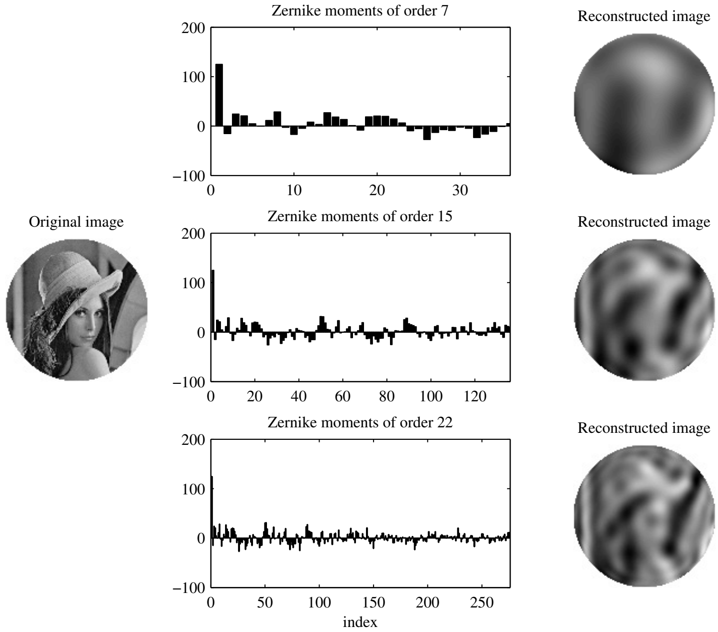

Fig. 1 Reconstruction of grayscale image Lena using Zernike moments of orders 7 (top), 15 (middle) and 22 (bottom). The original image is given (left) as well as the reconstructed images, for each analyzed order (right).

where $V^*_{nm}(x,y) = V_{n,-m}(x,y)$. In order to compute the Zernike moments of a digital image, we need to change integrals with summations:

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x,y) V^*_{nm}(x,y), \quad x^2 + y^2 \leqslant 1 \tag{4}$$

Assuming that all Zernike moments A_{nm} of $f(x,y)$ up to order N_Z are known, the image approximation can be obtained by:

$$f'(x,y) = \sum_{n=0}^{N_Z} \sum_m A_{nm} V_{nm}(x,y) \tag{5}$$

When computing the Zernike moments of a given image, the center of the image is taken as the origin and the pixel coordinates are mapped to the range of the unit circle. Those pixels falling outside the unit circle are not used in the computation. Figure 1 shows the reconstruction results for 128×128 grayscale image Lena with Zernike moments of orders up to $N_Z = 7$, 15 and 22. The total number of moments is 36, 136, and 276, respectively. We see that the lower order Zernike moments capture basic shape, while the higher order moments contain fine details of the image.

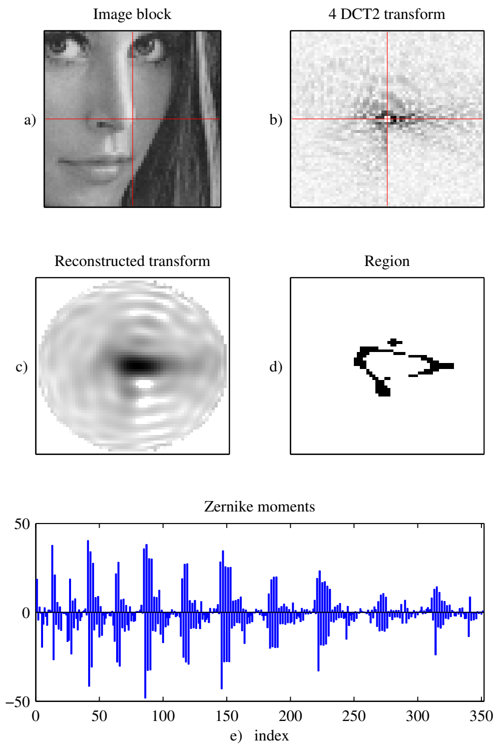

Fig. 2 Zernike moments of the DCT image obtained by dividing the original image into four blocks and rearranging 2D DCT of the blocks into centralized transform domain representation.

3 Zernike moments based selection of watermarking regions

The use of Zernike moments calculated for each image block in the transform domain is considered, in order to determine the blocks (regions) suitable for watermark embedding. The Zernike moments are calculated on a unit circle domain. Since the image blocks have a square shape, it is necessary to cut out parts outside the circle. In that sense, all coefficients of interest have to be within the unit circle close to the origin. This could be achieved using, for instance, the 2D discrete Fourier transform, since the low-frequency coefficients are concentrated in the center of the transformation domain, while high-frequency coefficients are negligible.

In the case of 2D DCT, the most significant coefficients are concentrated in the upper-left corner of the domain. Hence, to calculate Zernike moments for the DCT blocks, the following procedure is applied. The selected block is divided into four equal parts (Figure 2a), and the DCT is calculated for each part individually. Then, the different parts are rearranged in the single block, such that all significant coefficients are concentrated in the center of the plane, as shown in Figure 2b. The Zernike moments are calculated for this block, and they are shown in Figure 2e.

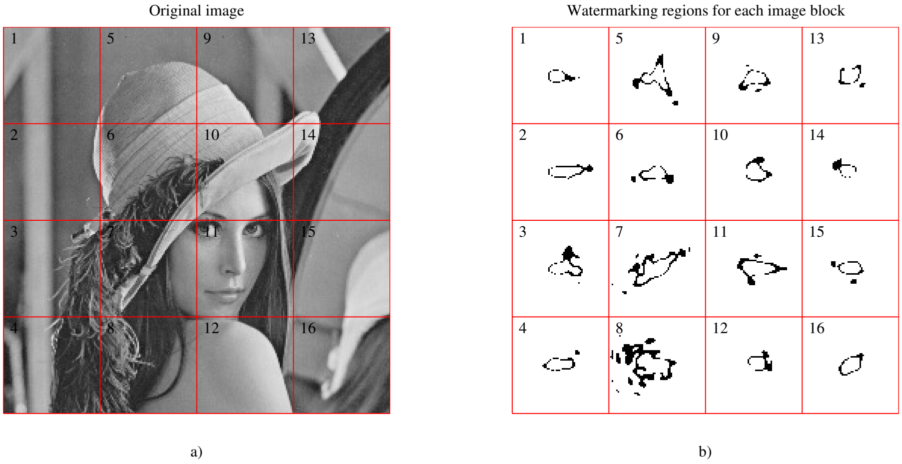

Fig. 3 a) The original image divided into blocks, b) Corresponding regions suitable for the watermark embedding.

The obtained moments are used to perform the reconstruction of the selected block and to obtain reconstructed transform-domain image $RT(x,y)$ as shown in Figure 2c. This reconstruction represents an approximation of DCT coefficients obtained using a finite number of Zernike moments. Here we use Zernike moments up to the 25^{th} order resulting in 351 moments.

To insert a watermark in the transformation domain, we select region R where the values of coefficients are not too large (to avoid visible image degradation) and must not be neither too small, because one could be able to easily remove and destroy the watermark without affecting image quality. Watermarking region is obtained from the transform domain image $RT(x,y)$ as:

$$\mathbb{R} = \left\{ (x,y) : \theta_1 < \frac{RT(x,y)}{\max\limits_{x,y} R(x,y)} < \theta_2 \right\} \tag{6}$$

where the thresholds θ_1 and θ_2 are determined empirically, as a percent of the maximum in the reconstructed transform. The region shown in Figure 2d was obtained for the following threshold values: $\theta_1 = 0.15$ and $\theta_2 = 0.25$. By repeating the above steps for each block of the image, we obtain the regions for watermark embedding, as shown in Figure 3. We may observe that each region contains different number of coefficients for watermarking. In the considered example (Figure 3), the total number of coefficients for all regions is 1582.

The proposed watermarking procedure based on Zernike moments and four rearranged 2D DCT (Figure 4) is summarized in the Algorithm 1.

Algorithm 1 Watermarking algorithm

Input:

Block/Image $\mathbf{A}_{N\times N}$ with elements $A(n,m)$, $n=1,2,\dots,N$, $m=1,2,\dots,N$, N is even

Maximal Zernike moments order N_Z

Thresholds θ_1 and θ_2

1: Divide image into four parts
$A_1(n,m)=A(n,m)$
$A_2(n,m)=A(n,\frac{N}{2}+m)$
$A_3(n,m)=A(\frac{N}{2}+n,m)$
$A_4(n,m)=A(\frac{N}{2}+n,\frac{N}{2}+m)$

2: Calculate 2D DCT for each part separately:
$\mathbf{T}_k=\mathrm{DCT}_{2\mathrm{D}}(\mathbf{A}_k)\ \ k=1,2,3,4$

3: Arrange 2D DCT matrices $\mathbf{T}_k$ into matrix $\mathbf{T}$ as illustrated in Figure 4. The coefficients of matrix $\mathbf{T}$ are:

$$T(n,m)=\begin{cases} T_1(\frac{N}{2}-n,\frac{N}{2}-m) \text{ for } n\leqslant\frac{N}{2} \text{ and } m\leqslant\frac{N}{2} \\ T_2(\frac{N}{2}-n,m-\frac{N}{2}) \text{ for } n\leqslant\frac{N}{2} \text{ and } m>\frac{N}{2} \\ T_3(n-\frac{N}{2},\frac{N}{2}-m) \text{ for } n>\frac{N}{2} \text{ and } m\leqslant\frac{N}{2} \\ T_4(n-\frac{N}{2},m-\frac{N}{2}) \text{ for } n>\frac{N}{2} \text{ and } m>\frac{N}{2} \end{cases}$$

4: Set middle coefficients to zero. These coefficients correspond to the zero frequency (DC) in 2D DCT.

$$T_0(n,m)=\begin{cases} 0 \text{ for } (n,m)\in\left\{\left(\frac{N}{2},\frac{N}{2}\right),\left(\frac{N}{2},\frac{N}{2}+1\right),\left(\frac{N}{2}+1,\frac{N}{2}\right),\left(\frac{N}{2}+1,\frac{N}{2}+1\right)\right\} \\ |T(n,m)| \ \ \text{otherwise} \end{cases}$$

5: Calculate Zernike moments of $\mathbf{T}_0$ up to the order N_Z

6: Reconstruct image from Zernike moments: $RT(n,m)$

7: Determine watermarking region $\mathbb{R}$ according to (6) using given thresholds θ_1 and θ_2

8: Embed watermark sequence in the coefficients $\mathbf{T}$ selected by region $\mathbb{R}$

9: Calculate inverse 2D DCT transform for each part of $\mathbf{T}$ and arrange them into image.

Output:

Watermarked image and watermarking region.

4 Block selection criterion

In the described approach, the watermark is embedded in each image block regardless the block features. Further improvement of the proposed procedure can be achieved by defining the criterion for the selection of blocks suitable for watermark embed-

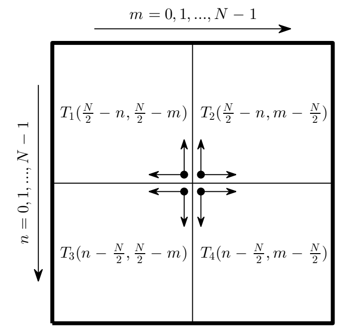

Fig. 4 Illustration of arranging four 2D DCT matrices.

ding. For example, for uniform blocks (flat image regions), the embedded watermark could be visible, and should not be used for watermarking. On the other hand, a block consisting of a large number of details (busy region) is a good candidate for watermarking.

We can make the decision on the level of details in the observed block by analysing its higher order Zernike moments. If these are small (negligible), then the block is not suitable for watermark embedding. This criterion can be mathematically written using the ℓ_1-norm of higher order Zernike moments ($n_L \leq n \leq n_H$) as follows:

$$\sum_{n=n_L}^{n_H} \sum_{m} |A_{nm}| < \Theta, \tag{7}$$

where Θ is a threshold. For the block selection procedure, we used higher order Zernike moments from $n_L = 4$ to $n_H = 7$ (26 coefficients). The regions selected according to this criterion for the image Lena, obtained using the threshold $\Theta = 200$, are shown in Figure 5. The total number of selected coefficients is 915. In Figure 6, the Zernike moments are represented for regions 5, 15, 7 and 11 from Figure 3. The moments used in equation (7) are marked in dark (red) colour. It is obvious that the regions in Figure 6a and 6b are not suitable for watermark embedding, while the regions shown in Figure 6c and 6d are good candidates for watermarking.

5 Experimental results

Example 1: The efficiency of the proposed approach regarding the imperceptibility and detectability is illustrated in this example. Several test images are observed, belonging to the standard MATLAB test-set for image processing (Lena, Cameraman, Baboon, Autumn, etc.). The illustrations are provided for test image Lena since there is no significant difference in performance for other test images. Therefore, Figure 7a presents the original 256×256 Lena image, while Figure 7b shows the watermarked image. Note that there are no visible degradations of the watermarked image. The PSNR and SSIM achieved for the watermarked image are 46.5 dB, and 0.9957 which also demonstrates a high image quality.

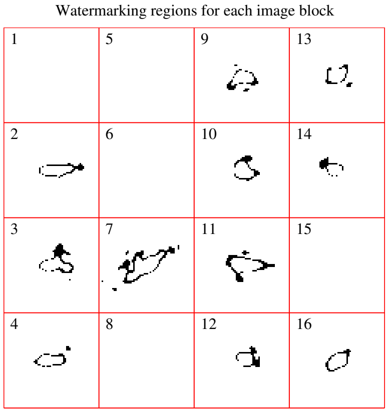

Fig. 5 Selected blocks and corresponding regions for the watermark.

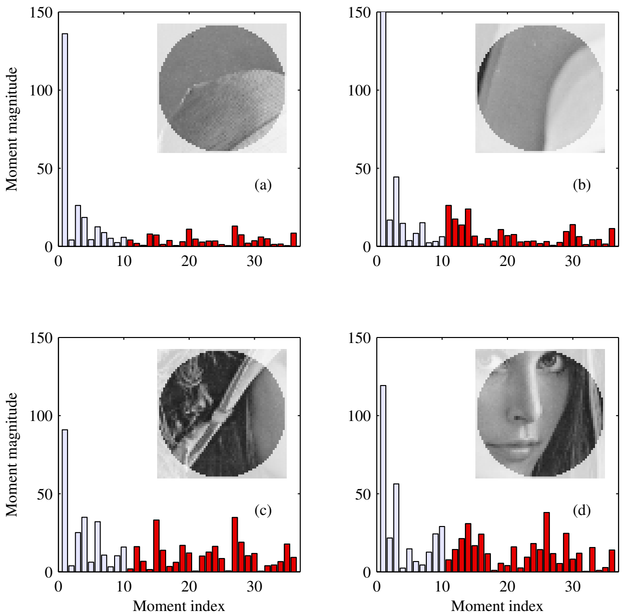

Fig. 6 Illustration of the criterion for blocks selection. Higher order moments used in (7) are given in red (dark) color.

For the purpose of comparison, we apply the standard DCT watermarking procedure, using 8×8 image blocks. By selecting three 2D DCT coefficients in each block the total number of coefficients selected for watermark embedding is 3072, which is almost twice the number of components used in the proposed approach. Namely, the idea is to test both procedures under the same PSNR and approximately the same number of components. However, in the case of standard 2D DCT watermarking it is difficult to achieve high PSNR (nonvisible degradations) and good detection with

Fig. 7 The result of the proposed watermark method: a) original image b) watermarked image.

Fig. 8 Standard DCT-based watermarking: a) Original image, b) Watermarked image.

the number of coefficients used in our approach. Therefore, the number of DCT coefficients in the standard watermarking is doubled. The original and watermarked image for the standard DCT-based watermarking procedure are shown in Figure 8a and 8b, respectively. Using this method, the achieved PSNR and SSIM of the watermarked image are 46.5 dB, and 0.9975. They are almost equal to the PSNR and SSIM achieved using the proposed approach (which is important for a fair comparison).

Furthermore, we compare the results of watermark detection using the standard correlation detector for the proposed Zernike-based DCT method and standard DCT watermarking procedure. Figure 9 shows the results of watermark detection (Figure 9a the proposed watermarking scheme, Figure 9b standard DCT watermarking). Although the number of watermarking coefficients for the proposed method

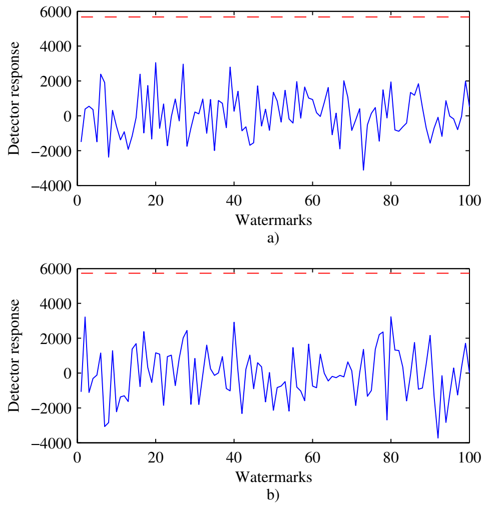

Fig. 9 Watermark detection: a) The proposed method, b) DCT method. 100 wrong keys (blue solid line) and one correct key – watermark (red dashed line) are used.

was twice smaller, the detection results are even better compared to the standard DCT approach.

Example 2: In this example, the performance of the proposed approach is tested in the presence of different attacks and compared to the standard DCT watermarking scheme. For the proposed scheme, the test images are divided into blocks and the blocks selection method is performed. For each selected block the watermarking region is determined. The total number of watermarking coefficients in all regions is 915. The same number of coefficients is used in the DCT watermarking by selecting 305 8×8 blocks out of the total 1024 blocks. Three coefficients are used for watermarking in each selected block. A set of attacks is applied after the watermarking and the watermark detection is examined. The output of correlation detector is calculated for 99 wrong keys and for one true key - watermark (placed on the position 50, marked with a point in Figure 10 and Figure 11. For easier comparison of the results, the horizontal lines are added, representing the three-sigma rule borders, numerically calculated for the results obtained by using wrong key sequences.

The following attacks are considered:

JPEG compression

The JPEG compression attack with quality levels 30 % and 10 % is analyzed as well. The results are presented in Figures 10a and 10b for the standard DCT watermarking, and Figures 11a and 11b for the proposed approach. Note that the propose procedure provides reliable detection unlike in the case of the standard DCT procedure.

Noise

Salt&pepper impulse noise with intensity 0.05 is considered (Figures 10c and 11c) as well as the multiplicative speckle noise with variance equal to 0.04 (Figures 10d and 11d). Again the reliable detection is achieved in the case of the proposed approach, while in the case of standard DCT watermarking procedure the detection fails.

Geometric distortions
Trimming a block of 100×100 pixel is considered. The sliced block is approximately 15 % of the whole image. Results are presented in Figures 10e and 11e. Note that none of the methods can provide detection in this case of attack. Further, the rotation for angle 30° is performed and the results are presented in Figures 10f and 11f for the two considered procedures. Note that the watermark detection under rotation attack is possible with the proposed method.

Median filter
Watermarked images are filtered using median filters of size 3×3 and 5×5. The filtered image and detector responses are given in Figures 10g and 10h (for the standard DCT approach) and Figures 11g and 11h (for the proposed approach). Comparing the obtained results, we may conclude that the watermark detection in the case of standard DCT procedure failed, while the proposed method would detect the watermark in the case of filter size 3×3, but also fail for the mask 5×5.

The proposed procedure, along with block DCT watermarking is repeated for various watermark intensities. The detector output in terms of PSNR and SSIM is calculated and presented in Figure 12. Random watermarking key is used in each realization for subplots (a) and (b). Results presented in subplots (c) and (d) correspond to the single watermark key used in each realization. The proposed method is presented in blue (dots for true keys and circles for wrong keys) and block DCT in red color (plus marks for true and cross marks for wrong keys). We can conclude that the proposed method achieve higher detector output for same PSNR or SSIM.

The proposed method is statistically tested on image “Lena” and 49 standard test images (available online at `http://decsai.ugr.es/cvg/CG/base.htm`). The ratio of the mean detector output for true keys and standard deviation of the detector output for wrong keys is used as a performance measure.

The comparison with standard watermarking methods [1,2] is done. The proposed method is compared with full 2D DCT watermarking (1035 low-frequency coefficients are omitted), and with 8×8 block DCT using the coefficients starting from the third diagonal. The wavelet transform based watermarking with haar (db1) wavelet (third level details are used as watermarking coefficients) is analyzed as well.

Test images are resized to 256×256 pixels. For each image and attack 100 random keys are analyzed. The watermark intensity is set to achieve PSNR of 40 dB in all analyzed cases. The results are presented in Table 1 for image “Lena” and 6 standard test images. The highest achieved ratio for each analyzed case is marked with bold number.

The mean values of the performance ratio for different attacks over 50 test images is presented in Table 2. It is obvious that the proposed method outperforms other analyzed methods in most cases. Namely, only for median 3×3 attacks, the proposed method is not the best one.

The calculation time for proposed watermarking procedure is compared with classical DCT based watermarking, Zernike calculation method proposed in [19] and watermarking procedure proposed in [22]. The results are given in table 3. In all considered cases 256×256 gray-scale image is used. The calculation time for the proposed method is measured on Intel-based Pentium IV PC, at 3.2GHz, with 2GB RAM, MATLAB version R2012b. The experiments reported in [19,21] are performed with

Table 1 Comparison of the proposed method with standard watermarking techniques [1,2]. The ratio of the detector output mean (for true keys) and standard deviation (for wrong keys) is presented.

Attack	Image	Proposed	DCT full	DCT block	Wavelet
no attack	Lena	**7.09**	4.59	3.97	4.06
	Test image #1	**7.78**	5.03	4.40	5.10
	Test image #2	**9.58**	6.50	5.94	6.31
	Test image #3	**8.43**	6.28	3.68	4.54
	Test image #4	**11.64**	10.44	7.46	7.08
	Test image #5	**8.62**	6.72	6.77	5.81
	Test image #6	**11.97**	9.32	6.93	6.14
jpeg Q=30	Lena	**7.09**	4.69	4.34	3.67
	Test image #1	**6.35**	5.57	4.59	4.22
	Test image #2	**9.49**	6.13	5.81	6.16
	Test image #3	**7.72**	5.66	4.15	4.13
	Test image #4	**11.92**	9.92	7.24	5.88
	Test image #5	**8.03**	7.29	5.72	6.34
	Test image #6	**10.58**	8.33	8.22	5.81
salt & pepper	Lena	**6.40**	4.40	4.13	3.99
	Test image #1	**6.80**	5.35	3.92	3.76
	Test image #2	**8.53**	6.08	4.79	5.75
	Test image #3	**7.73**	5.94	3.91	4.10
	Test image #4	**9.93**	8.61	7.21	6.20
	Test image #5	**7.62**	5.85	5.76	5.57
	Test image #6	**8.18**	8.17	5.82	5.60
crop	Lena	**6.06**	4.57	2.79	3.74
	Test image #1	**6.49**	4.73	3.73	3.79
	Test image #2	**7.49**	4.46	3.70	4.23
	Test image #3	**7.67**	4.79	2.87	3.02
	Test image #4	**11.86**	6.89	5.14	4.60
	Test image #5	**7.56**	4.79	4.50	4.47
	Test image #6	7.33	**7.59**	4.94	3.67
rotation 30deg	Lena	**5.80**	3.98	3.51	2.96
	Test image #1	**6.10**	5.18	3.22	3.99
	Test image #2	**10.19**	5.06	3.87	4.66
	Test image #3	**6.02**	5.56	3.55	3.32
	Test image #4	**9.74**	9.37	4.72	5.95
	Test image #5	**7.16**	5.83	4.35	4.51
	Test image #6	**10.00**	7.91	6.18	4.88
median 3x3	Lena	**6.11**	4.76	3.20	3.24
	Test image #1	**5.60**	5.39	3.35	3.18
	Test image #2	**8.80**	6.73	3.83	4.25
	Test image #3	5.48	**6.34**	2.87	3.74
	Test image #4	**10.29**	9.54	6.24	6.60
	Test image #5	**8.50**	6.58	5.31	4.61
	Test image #6	7.14	**9.41**	4.89	3.59

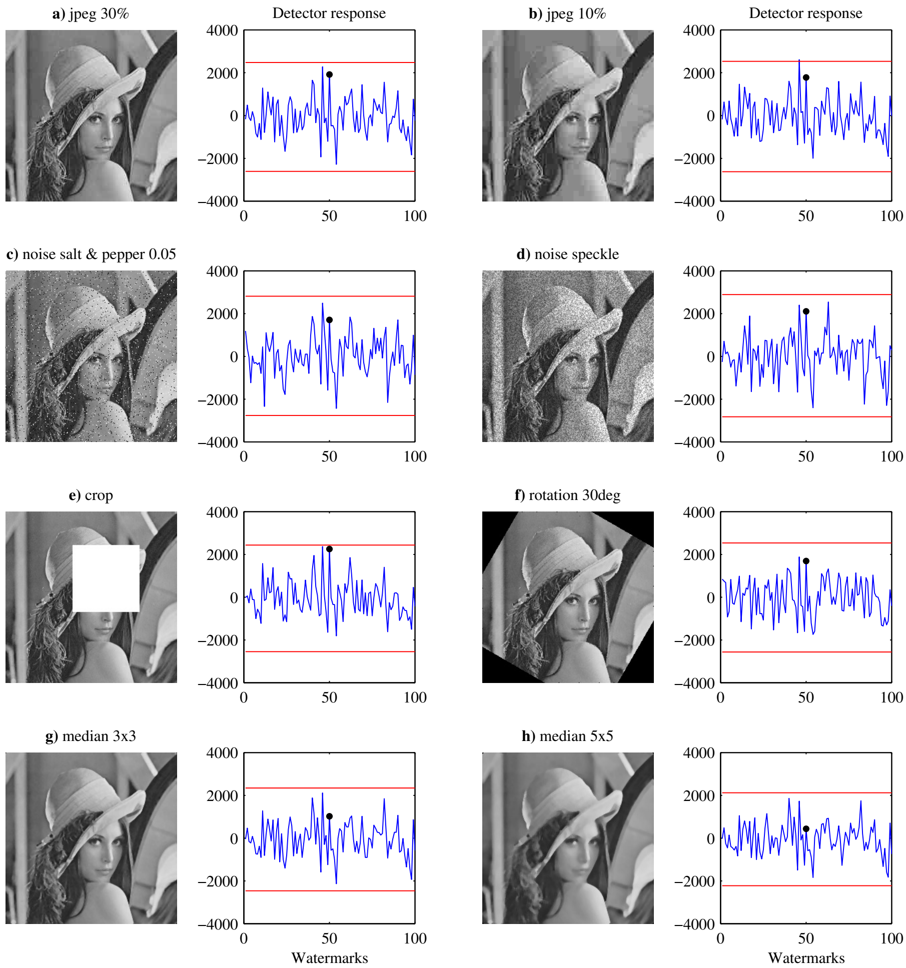

Fig. 10 Detection results for standard DCT watermarking under attacks: a) JPEG–30% quality, b) JPEG–10% quality c) salt&pepper noise, d) speckle noise, e) trimming 15% of the image, f) 30° rotation, g) median filter 3×3, h) median filter 5×5. Watermarking is performed in 915 coefficients with 46.5 dB.

Table 2 Average performance ratio of the analyzed method for various attacks over 50 test images.

Attack	Proposed	DCT full	DCT block	Wavelet
no attack	9.63	7.26	5.69	5.34
jpeg Q=30	8.39	7.09	5.66	4.86
salt & pepper	7.96	6.40	5.03	4.65
crop	7.49	5.76	3.97	3.73
rotation 30deg	7.99	5.92	4.35	4.15
median 3x3	6.31	6.89	4.30	4.00
Average	7.96	6.55	4.83	4.46

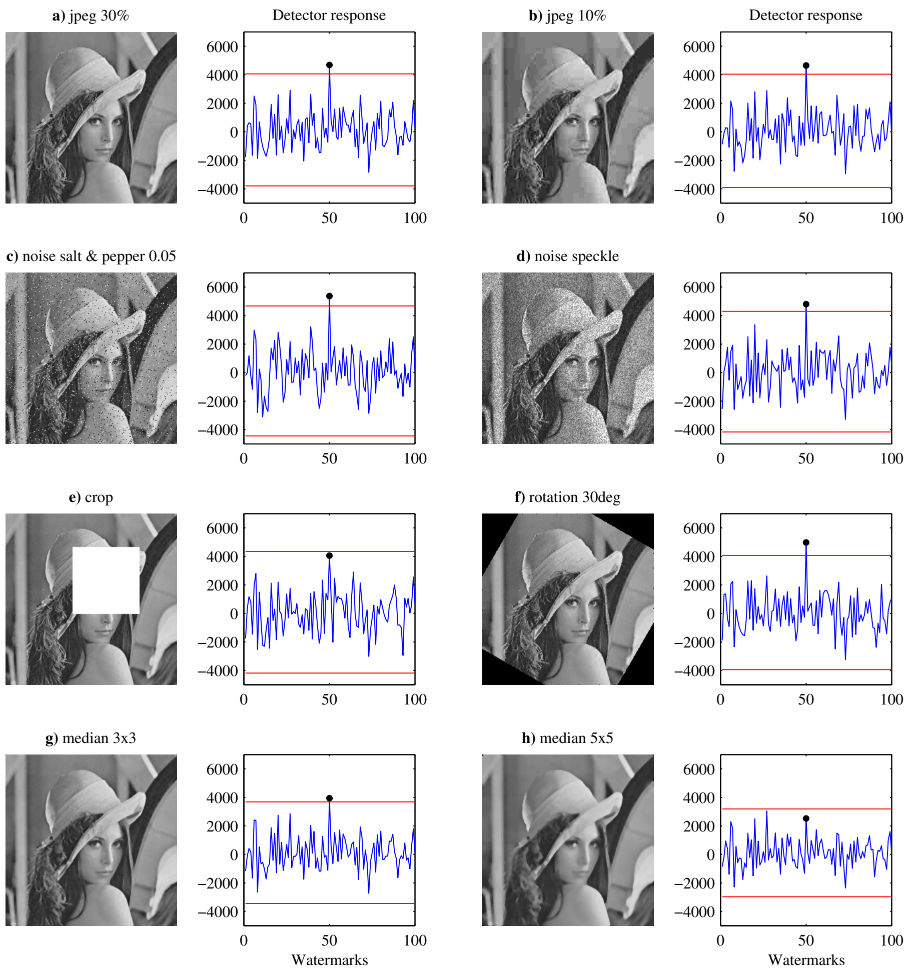

Fig. 11 Detection results for the proposed procedure under attacks: a) JPEG–30% quality, b) JPEG–10% quality c) salt&pepper noise, d) speckle noise, e) trimming 15% of the image, f) 30° rotation, g) median filter 3×3, h) median filter 5×5. Watermarking is performed in 915 coefficients with 46.5 dB.

MS Visual C++ 6.0 and MATLAB (respectively) on similar PC configuration. High computational load of the higher order Zernike moments is reported in [21] as well.

The proposed procedure increases calculation complexity, compared to the classical DCT watermarking, due to Zernike moments and specific 4 DCT blocks calculation (as illustrated in Figure 4). In the proposed method we use only the lower order (7 and 25) Zernike moments calculated for 64×64 image blocks, unlike the watermarking procedures based on spatial-domain Zernike moments. Note that, the method proposed in [19] is for Zernike moments calculation only, that is only part of the watermarking procedure.

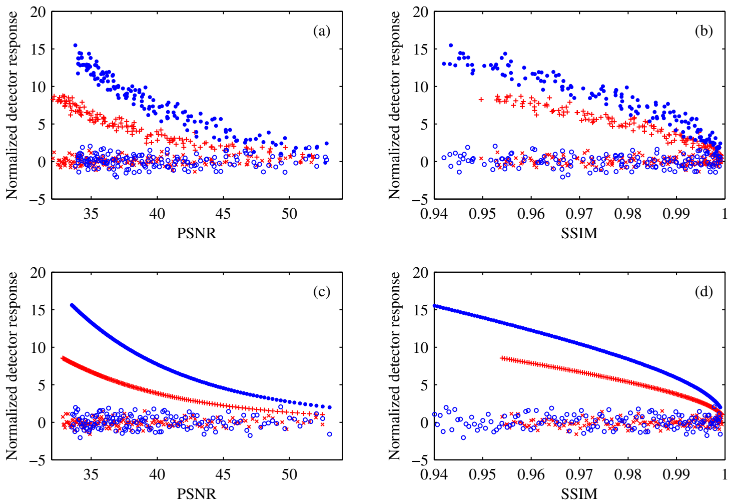

Fig. 12 Normalized detector output in terms of PSNR and SSIM for random watermarking keys (a,b) and for the single watermarking key (c,d). The proposed method is presented in blue (dots for true and circles for wrong keys) and block DCT watermarking in red color (plus marks for true and crosses for wrong keys).

Table 3 Calculation complexity comparison

Method	Calculation time
The proposed method – watermark embedding	5.85 s
The proposed method – watermark detection	1.17 s
Classical DCT watermarking – watermark embedding	0.73 s
Classical DCT watermarking – watermark detection	0.32 s
Method [19] – Zernike moments of order 24 calculation	0.92 s
Method [22] – watermark embedding	60 s

6 Conclusion

The proposed algorithm uses transform domain Zernike moments for image watermarking. The Zernike moments are calculated for the image blocks in the rearranged 2D DCT domain. By applying two thresholds on the reconstructed transformation, we obtain the region suitable for watermarking. The size of the regions depends on the chosen thresholds and the level of details in the considered image block. The blocks with no significant information have been omitted during watermark embedding. The selection of blocks suitable for watermarking is done based on the analysis of the ℓ_1-norm of higher order Zernike moments. The experimental results show that the proposed method provides better robustness to the common attacks (median filtering, additive noises, image compression, and geometric distortions) in comparison with the standard watermarking approaches.

Acknowledgements This work is supported by the Montenegrin Ministry of Science, project grant funded by the World Bank loan: CS-ICT "New ICT Compressive sensing based trends applied to: multimedia, biomedicine and communications".

References

1. M. Barni, F. Bartolini, Watermarking Systems Engineering, New York: Marcel Dekker, Inc. (2004)
2. S. Stanković, I. Orović, and E. Sejdić, Multimedia Signals and Systems: Basic and Advance Algorithms for Signal Processing, Springer-Verlag, New York, (2015)
3. E. Muharemagić, B. Furht, Survey of Watermarking Techniques and Applications, CH. 3 in Multimedia Watermarking Techniques and Applications, B. Furht and D. Kirovski, editor, Auerbach Publication, 91–130, (2006)
4. Chuan Qin, Chin-Chen Chang, and Tai-Jung Hsu, Reversible data hiding scheme based on exploiting modification direction with two steganographic images, Multimedia Tools and Applications, **74**(15), 5861-5872, (2015)
5. Chuan Qin, Xinpeng Zhang, Effective reversible data hiding in encrypted image with privacy protection for image content, Journal of Visual Communication and Image Representation, **31**, 154-164, (2015)
6. Xinpeng Zhang, Zhenxing Qian, Yanli Ren, Guorui Feng, Watermarking with flexible self-recovery quality based on compressive sensing and compositive reconstruction, IEEE Transactions on Information Forensics and Security, **6**(4), 1223-1232, (2011)
7. Chuan Qin, Chin-Chen Chang, Yi-Ping Chiu, A novel joint data-hiding and compression scheme based on SMVQ and image inpainting, IEEE Transactions on Image Processing, **23**(3), 969-978, (2014)
8. A. Nikolaidis and I. Pitas, Asymptotically optimal detection for additive watermarking in the DCT and DWT domains, IEEE Transaction on Image Processing, **12**(5), 563–571, (2003)
9. J. R. Hernandez, M. Amado, and F. Perez Gonzalez, DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure, IEEE Transaction on Image Processing, **9**(1), 55–68, (2000)
10. D. K. Tsolis, S. Sioutas, T. S. Papatheodorou, A multimedia application for watermarking digital images based on a content based image retrieval technique, Multimed Tools and Applications, **47**(3), 581-597, (2010)
11. I. Orovic, M. Orlandic, S. Stankovic, An Image Watermarking Based on the pdf Modeling and Quantization Effects in the Wavelet Domain, Multimedia Tools and Applications, **70**(3), 1503–1519 (2014)
12. M. Vincent, M. Chabert, B. Lacaze, A Spread Spectrum Watermarking Scheme Based on Periodic Clock Changes for Digital Images, In Proceedings of Information Hiding, **3727**, 91–105, (2005)
13. J. Liu, K. She, H. Wu, Blind Image Watermarking using Dual Embedding Scheme in the Wavelet Transform Domain, Journal of Computational Information Systems **6**(6), 1887–1896, (2010)
14. S. Stankovic, I. Orovic, M. Chabert, B. Mobasseri, Image Watermarking based on the Space/Spatial-Frequency Analysis and Hermite Functions Expansion, Journal of Electronic Imaging, **22**(1), 013014, (2013)
15. S. Stankovic, I. Orovic, N. Zaric, C. Ioana, Two Dimensional Time-Frequency Analysis based Eigenvalue Decomposition Applied to Image Watermarking, Multimedia Tools and Applications, **49**(3), 529–543, (2010)
16. J. Flusser, T. Suk., B. Zitová, Moments and Moment Invariants in Pattern Recognition, John Wiley&Sons, (2009)
17. A. Tahmasbi, S. Fatemeh, S. B. Shahriar, Classification of benign and malignant masses based on Zernike moments, Computers in biology and medicine, **41**(8), 726–735, (2011)
18. L. Kotoulas, I. Andreadis, Accurate Calculation of Image Moments, IEEE Transactions on Image Processing, **16**(8), 2028–2037, (2007)
19. Sun-Kyoo Hwang, Whoi-Yul Kim, A novel approach to the fast computation of Zernike moments, Pattern Recognition, **39**(11), 2065–2076, (2006)
20. G. A. Papakostas, Y. S. Boutalis, C.N. Papaodysseus, D.K. Fragoulis, Numerical error analysis in Zernike moments computation, Image and Vision Computing, **24**(9), 960–969, (2006)
21. Q. Chen, X. Yang, J. Zhao, Robust image watermarking with Zernike moments, in: Canadian Conference on Electrical and Computer Engineering, Saskatoon, SASK., 1340-1343, (2005)

22. Hyung Shin Kim, and Heung-Kyu Lee, Invariant image watermark using Zernike moments, IEEE Transactions on Circuits and Systems for Video Technology, **13**(8), 766–775, (2003)
23. Nitin Singhal, Young-Yoon Lee, Chang-Su Kim, Sang-Uk Lee, Robust image watermarking using local Zernike moments, Journal of Visual Communication and Image Representation, **20**(6), 408–419, (2009)
24. Say Wei Foo, Qi Dong, A normalization-based robust watermarking scheme using Zernike moments, World Academy of Science, Engineering and Technology, **3**(11) 1968–1973, (2009)