

Hiding Correlation-Based Watermark Templates using Secret Modulation

Jeroen Lichtenauer, Iwan Setyawan, Reginald Lagendijk

Information and Communication Theory Group, Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands

ABSTRACT

A possible solution to the difficult problem of geometrical distortion of watermarked images in a blind watermarking scenario is to use a template grid in the autocorrelation function. However, the important drawback of this method is that the watermark itself can be estimated and subtracted, or the peaks in the Fourier magnitude spectrum can be removed. A recently proposed solution is to modulate the watermark with a pattern derived from the image content and a secret key. This effectively hides the watermark pattern, making malicious attacks much more difficult. However, the algorithm to compute the modulation pattern is computationally intensive. We propose an efficient implementation, using frequency domain filtering, to make this hiding method more practical. Furthermore, we evaluate the performance of different kinds of modulation patterns. We present experimental results showing the influence of template hiding on detection and payload extraction performance. The results also show that modulating the ACF based watermark improves detection performance when the modulation signal can be retrieved sufficiently accurately. Modulation signals with small average periods between zero crossings provide the most watermark detection improvement. Using these signals, the detector can also make the most errors in retrieving the modulation signal until the detection performance drops below the performance of the watermarking method without modulation.

Keywords: Image watermarking, geometrical distortion, autocorrelation template, hiding, modulation

1. INTRODUCTION

Digital watermarking is the process of embedding information within digital media itself, imperceptible to the ‘user’ of the media under normal circumstances. A robust digital watermark must survive all signal processing that can be expected in its application. In some applications, users might try to deliberately remove the watermark. Such a malicious attack can be any kind of distortion. The only limitation to the distortion is that it does not eliminate the commercial value of the work. The far most challenging problem in distortion robustness of digital watermarks, at this moment, is robustness to de-synchronization. In image and video watermarking, even a very small geometrical distortion, like a translation over a few image pixels, cropping of the image or column/line removal, already confuses the watermark detector completely. Also more complex geometrical transformations can be applied to an image without affecting the perceptible quality. Figure 1 shows a Random Bending Attack (RBA), which is part of the widely used watermark benchmark application called “Stirmark”. The small bending applied to the image is imperceptible but disastrous for nearly all the currently used watermarking methods.



Figure 1. Example of the Random Bending Attack (RBA). Image (b) shows the result of a small geometrical bending of the ‘Lena’ image (a).

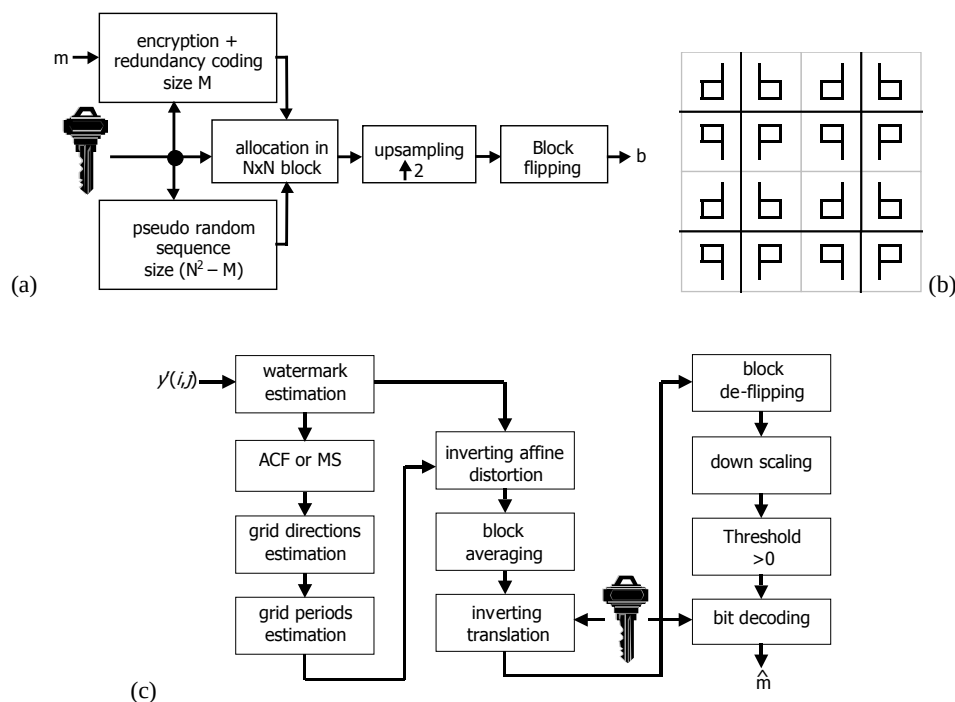


Figure 2. Watermark embedding (a) and detection procedure (c). At embedding the encrypted and encoded message m is allocated in M pixels in the $N \times N$ block according to a key. The remaining $N \times N - M$ pixels are filled with a pseudo random sequence. The block is upsampled and flipped horizontally and vertically to obtain the watermark macro-block b of size $4N \times 4N$. The tiled watermark (b) shows the symmetry of the macro-block introduced by flipping.

Research is being conducted on methods that can solve this difficult problem. For watermarking in video sequences, the time axis can be used to obtain watermarking methods that are robust to geometrical distortion. This kind of watermarking is already in an advanced state of research, for instance by Niu et. al⁽¹⁾ and Van Leest et. al⁽²⁾. For still images, one branch of countermeasure research attempts to develop better image-watermarking techniques that are robust against different types of geometrical transforms^(3,4). A complementary branch of countermeasures focuses on inverting the geometrical transform prior to watermark detection, to avoid confusion of the watermark detector. The objective of the inversion of geometrical distortions is to restore as closely as possible the original (image) grid on which the watermark was embedded. In this case, reference information is required about the original undistorted pixel grid. This grid reference information is usually either the original non-watermarked image - leading to non-blind approaches⁽⁵⁾ -, or embedded structure or marker information - leading to blind inversion approaches^(6,7,8). A different blind approach is to perform a geometrical search for the watermark, however this has important drawbacks concerning false positive detection probability, as we explained in⁽⁹⁾.

Of the blind inversion approaches, the method proposed by Voloshynovskiy, Deguillaume and Pun^(7,8) shows the most potential. The authors use the autocorrelation function (ACF) of a tiled watermark block to estimate affine distortions in order to invert it prior to watermark detection. It is based on the method previously introduced in the work of M. Kutter⁽¹⁰⁾. We shall refer to this method as 'ACF watermarking'. The tiled watermark block in^(7,8), which is called the 'macro-block' b , is not only used for affine transform estimation, but the main part actually consists of the (encoded) watermark message bits. The block diagram of this approach is shown in Figure 2. A smaller part of the block contains a pseudo random sequence that is known by the watermark detector. This 'pilot signal' is necessary to find the orientation and/or translation of the watermark blocks. It can also be used to perform a correlation search for the watermark pattern.

The horizontally and vertically tiled pattern causes a grid of points in the two-dimensional ACF and the Fourier Magnitude Spectrum (MS) of the watermarked image. The grid can be represented by two vectors $\mathbf{u}$ and $\mathbf{v}$, as denoted in Figure 3. The directions of the vectors correspond to the horizontal and vertical grid directions respectively (perpendicular in the frequency domain) and their magnitudes correspond to the size of the watermark block in the

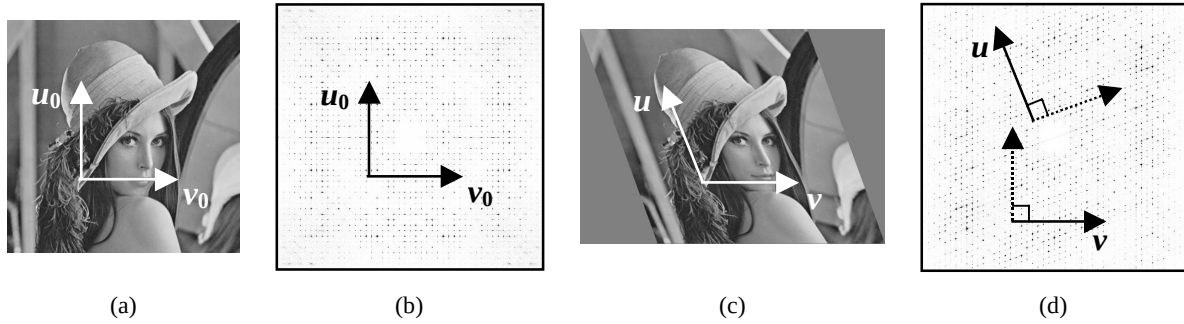


Figure 3. Example of the result of an affine transformation on the grid of the repeated watermark in the MS. The image (a) is sheared 20° (c). The grid in the MS (b) is also transformed in (d), but its directions, shown by the dotted arrows in are perpendicular to the directions in the spatial domain and in the ACF, shown by the solid arrows.

vector directions (inverse proportional to the period in the frequency domain). The vectors change proportionally to the affine transformation that is applied to the watermarked image. Because small watermark blocks are used, their geometrical distortion can be approximated by a local affine transformation, even after a (mild) non-linear transformation is applied to the image. When as many as possible macro-blocks are recovered (locally) in this way, the unknown data part of the tiled macro-block can be estimated by averaging the data bits over the blocks.

The most important drawback of ACF watermarking lies in its security: the regular peak grids in the ACF and MS are accessible without any knowledge of secure information (typically a key). Key knowledge is necessary only for finding the orientation of the estimated watermark block and decoding of the message. Without key knowledge the watermark block can still be estimated. Embedding the inverted pattern in the watermarked image will completely remove the watermark if the same visual masking is applied that was used for embedding the watermark. Delannay and Macq⁽¹¹⁾ have proposed to solve this problem by removing the periodicity of the watermark with a secret non-periodic modulation signal $s(i,j,k)$:

$$y(i, j) = x(i, j) + p(i, j)s(i, j, k), \quad (1)$$

where $y(i,j)$ is the watermarked image luminance at coordinates (i,j) , $p(i,j)$ is the tiled watermark pattern and $s(i,j,k)$ is the modulation signal that depends on a secret key k . The function $s(i,j,k)$ must have certain specific properties: 1) The essential property of the used watermark signal is its polarity, therefore, $s(i,j,k)$ must also be a binary function, i.e. $s(i, j, k) \in \{-1, 1\}$. 2) The result of function $s(i,j,k)$ must be secret. Without knowledge of key k , given only the watermarked image $y(i,j)$, the uncertainty about $s(i,j,k)$ must be maximal. 3) The signal $s(i,j,k)$ must always be registered with the watermark. However, a possible geometrical distortion cannot be known by the blind detector before-hand. Hence, the only solution is that $s(i,j,k)$ must be a function of both k and the image content. 4) Furthermore, $s(i,j,k)$ must be approximately the same for the original image, the watermarked image, and the (geometrically) distorted watermarked image. Thus,

$$s(i, j, k | x) = s(i, j, k | y) = s(i', j', k | y'), \quad (2)$$

where (i', j') is the location in a distorted watermarked image $y'(i', j')$, corresponding to the location (i, j) in $y(i, j)$. In other words, the modulation signal $s(i,j,k)$ must be calculated using a function that is insensitive to (watermarking) noise, scale, translation, rotation and shearing (i.e. affine transformation). In ⁽¹¹⁾ an algorithm to compute $s(i,j,k)$ is presented that satisfies most of the requirements mentioned above. In their method, a binary modulation signal is computed from the image content using a secret key. The authors show the robustness of the modulation function to different kinds of image distortion. However, they do not show the effect of actually applying the modulation function on any watermark template.

In this paper, we will apply the secret modulation function of ⁽¹¹⁾ to ACF watermarking. We will evaluate the influence of template modulation on watermark detection performance, and furthermore, show the result of errors in the modulation signal. In Section 2 the modulation function is described in more detail and also some modifications are proposed to make it more robust and computationally efficient. In Section 3 experimental results are presented showing

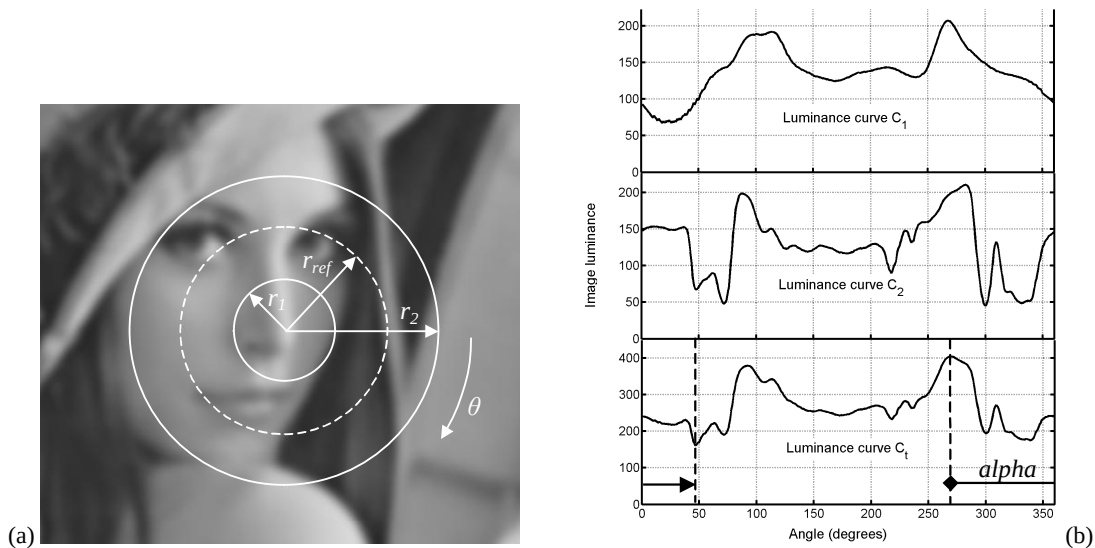


Figure 4. Example of determining α for an image pixel. (a) shows the reference circle with radius r_{ref} around a pixel on the nose of the image ‘Lena’ and two new circles C_1 and C_2 with a radii r_1 and r_2 determined by r_{ref} multiplied with k_1 and k_2 respectively. In (b) values of the image luminance over the circles C_1 and C_2 are plotted, adding these two curves gives C_t . α is calculated as the clock-wise distance from the maximum to the minimum of C_t .

the robustness of the modified hiding function and the influence of template hiding on watermark detection performance of ACF watermarking. Finally, in Section 4 our conclusions and recommendations for future work are presented.

2. SECRET MODULATION FUNCTION

In Section 2.1 the algorithm from⁽¹¹⁾ is described briefly. In Section 2.2. we present a modification to make the method more robust to image cropping and in Section 2.3. we explain how to make the algorithm more computationally efficient.

2.1. Original algorithm

Scale invariance is obtained by determining a reference radius r_{ref} for each image pixel using the local image content. The reference radius is defined as the radius of the smallest circle for which the mean luminance over its perimeter equals the mean luminance over its surface:

$$M_{disk}(i, j; r) = \frac{1}{\pi r^2} \int_0^r \int_0^{2\pi} x(i, j; r, \theta) r d\theta dr, \quad (3)$$

$$M_{circle}(i, j; r) = \frac{1}{2\pi r} \int_0^{2\pi} x(i, j; r, \theta) r d\theta, \quad (4)$$

where $x(i, j; r, \theta)$ is a polar representation of x at (i, j) ,

$$L(i, j; r) = M_{disk}(i, j; r) - M_{circle}(i, j; r), \quad r_{ref} = \arg \min_r L(i, j; r) = 0. \quad (5)$$

The reference radius is used to define two new circles around the pixel being considered, with radii r_1 and r_2 based on secret keys k_1 and k_2 respectively:

$$r_1 = k_1 * r_{ref}, \quad r_2 = k_2 * r_{ref} \quad (6)$$

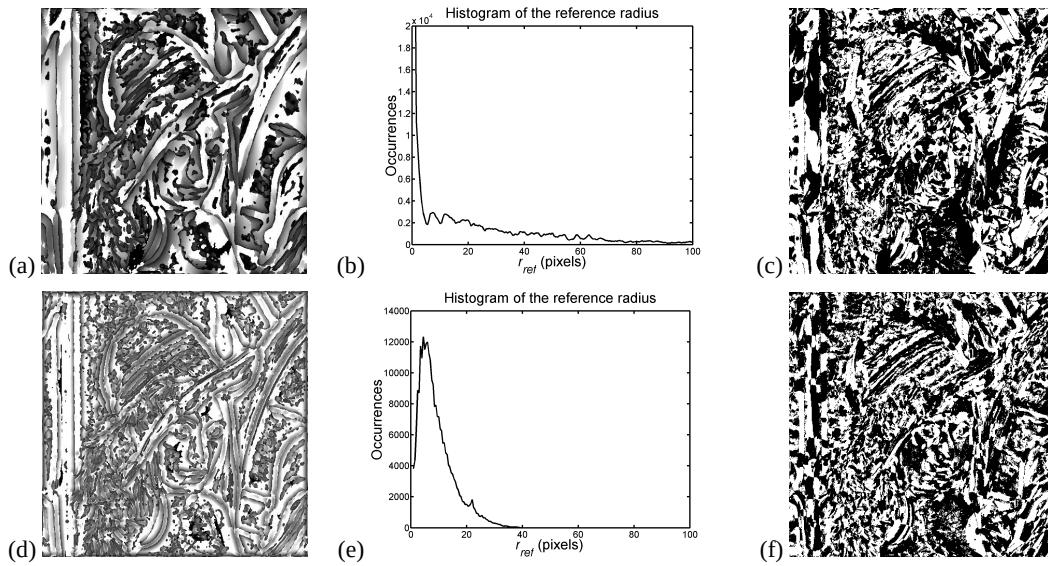


Figure 5. Pictures (a),(d) and histograms (b),(e) of r_{ref} for each pixel of the ‘Lena’ image and two examples of corresponding binary modulation signals (c),(f). The upper row using zero crossings of $L(r)$, the bottom row using zero crossings of its derivative.

For these two new circles, the image luminance curves $C_1(\theta)$ and $C_2(\theta)$ along the perimeters of the circles with radii r_1 and r_2 respectively are measured and added to obtain a new curve $C_t(\theta)$. This is illustrated in Figure 4. Now α is calculated as the clock-wise-directed angular distance from the maximum to the minimum of $C_t(\theta)$. The binary value of the pixel $s(i,j,k)$ is determined by thresholding α :

$$s(i, j, k) = \begin{cases} 1, & \alpha > \pi \\ -1, & \alpha \leq \pi \end{cases} \quad (7)$$

Instead of secretly varying k_1 and k_2 as keys, another method can be used to obtain key dependency. The authors of ⁽¹¹⁾ also suggest generating a pseudo random curve, using a key, which can be added to $C_t(\theta)$ before finding the minimum. This secret curve must be symmetrical around the maximum of $C_t(\theta)$, to ensure an equal probability for finding a positive and a negative value of $s(i,j,k)$.

The algorithm has some drawbacks that make it less effective for robust watermark hiding. First of all, to obtain robustness against noise, the image must be filtered with a low-pass filter, which introduces dependence to scaling. Second, an area around each pixel is used for computation of the modulation value. Consequently, when an image is cropped, a number of pixels near the new border are affected by this change in the pixel neighborhood. Since r_{ref} can become quite large, a lot of pixels are affected by cropping. A third drawback is the computational complexity of the algorithm. Especially the calculation of r_{ref} requires an enormous amount of computation. For each single pixel of the image the mean pixel values over circles and disks over a range of radii must be calculated using interpolation. This can be done using an efficient spline interpolation algorithm, but still our implementation in MATLABTM was very slow. Computation of r_{ref} for an image of 512x512 pixels took approximately 8 hours using a 2.4 Gigahertz processor. Improvements for the latter two drawbacks (robustness to cropping and computational efficiency) are presented in the next two subsections.

2.2. Improving robustness to cropping

The expected value of r_{ref} is quite large, in the order of 50-100 pixels, see Figure 5 (b). The large scale of this method has a negative effect on the robustness against cropping, because the border of the image affects all pixel values of $s(i,j,k)$ for which the radius of one of the used circles exceeds the pixel distance to the border.

One possibility to reduce the problem of cropping is to reduce the average size of r_{ref} by reducing the scale of the low-pass filter. However, this increases sensitivity to noise. Therefore, we propose to reduce the average size of r_{ref} by using the derivative of $L(r)$ in (5):

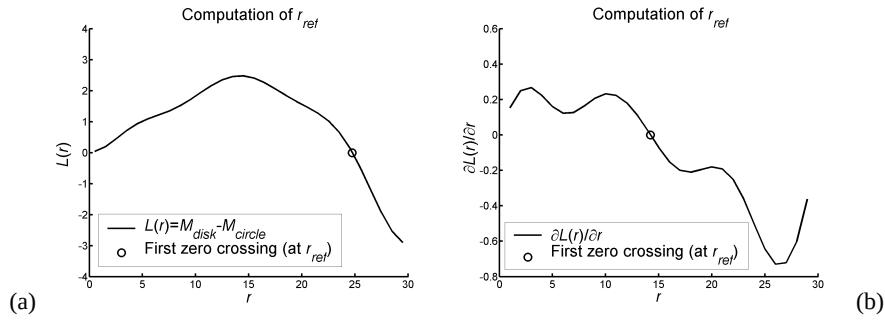


Figure 6. $L(r)$ in (a) and its derivative in (b) for different values of r , for pixel (50,100) of the ‘Lena’ image.

$$r_{ref} = \arg \frac{\partial}{\partial r} L(i, j; r) = 0. \quad (8)$$

In contrast to $L(r)$, its derivative does not have to start at zero, which reduces zero crossings at very low radii. But at the same time, the derivative has more zero crossings, which results in less extreme reference values. Figure 6 gives an example of functions $L(r)$ and $\partial L(r)/\partial r$. Because of more frequent zero crossings the reference radius will have a smaller expected value and also a smaller variance, which can be seen by comparing the histograms in Figure 5 (b) and (e) of the reference radii shown in (a) and (d). This will result in more robustness to cropping.

2.3. Reducing computational complexity

To reduce computational complexity of the modulation function, we propose a much more efficient implementation for computing $L(r)$, using Fourier-domain filtering. Computation of mean values over circles and disks around each pixel is equivalent to filtering with a circular and disk shaped filter respectively. Hence, $L(r)$ could be computed for all pixels at once by multiplying the DFT of the image, $\hat{F}_x(\omega_1, \omega_2)$, with the DFT of a disk with radius r , $\hat{F}_{disk}(\omega_1, \omega_2; r)$, minus the DFT of a circle with radius r , $\hat{F}_{circle}(\omega_1, \omega_2; r)$:

$$L(r) = \hat{F}^{-1} \left(\hat{F}_x(\omega_1, \omega_2) \times [\hat{F}_{disk}(\omega_1, \omega_2; r) - \hat{F}_{circle}(\omega_1, \omega_2; r)] \right). \quad (9)$$

The problem is that the circles and disks with small radii suffer a lot from aliasing when they are generated in the spatial domain, which makes their DFT unsuitable for this application. A different approach is to generate the DFT representations of the circles and disks directly in the DFT domain. The continuous Fourier transform of a perfect circle is a radial cosine function, while the transform of a disk is a radial sinc function. Unfortunately, the perfect circle and disk are IIR filters, and the discrete spectrum is periodic. Hence, the cosine and sinc functions corresponding to multiples of the sample frequency will overlap each other. Furthermore, the two-dimensional discrete Fourier domain is

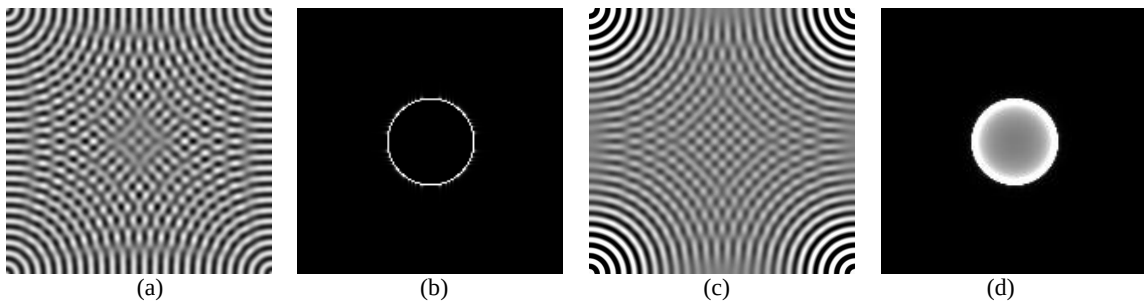


Figure 7. Shape generation in the DFT domain. On the left side a circle is generated, and on the right side a disk, both with a radius of 25 pixels. The frequency spectra generated by windowing the cosine and sinc functions are shown in (a) and (c) respectively. The results of their inverse DFT, which are the kernels used to approximate the integral equations (3) and (4), are shown in (b) and (d) respectively.

not rotational symmetrical. Simply applying a one-dimensional filter radially will introduce angle dependency. These problems are reduced by weighting the sampled continuous filter spectra between zero and the sampling frequency ω_s with an appropriate window function $W(\omega_1, \omega_2)$. We used the Bartlett window transformed to a 2-dimensional window using the McClellan function⁽¹²⁾:

$$W(\omega_1, \omega_2) = 1 - \frac{1}{\pi} \arccos \left\{ \frac{1}{2} \left(-1 + \cos\left(\frac{\omega_1}{\omega_s} \pi\right) + \cos\left(\frac{\omega_2}{\omega_s} \pi\right) + \cos\left(\frac{\omega_1}{\omega_s} \pi\right) \cos\left(\frac{\omega_2}{\omega_s} \pi\right) \right) \right\}. \quad (10)$$

Then, the spectra can be approximated as follows:

$$\hat{F}_{circle}(\omega_1, \omega_2; r) = \cos(2\pi \frac{\sqrt{\omega_1^2 + \omega_2^2}}{\omega_s}) W(\omega_1, \omega_2), \quad (11)$$

$$\hat{F}_{disk}(\omega_1, \omega_2; r) = \text{sinc}(2r \frac{\sqrt{\omega_1^2 + \omega_2^2}}{\omega_s}) W(\omega_1, \omega_2). \quad (12)$$

Aliasing is introduced on purpose, as this also occurs with the circle and disk kernels in the sampled image. Examples are shown Figure 7 for a circle and a disk with radii of 25 pixels in. The inverse DFT of the generated circle spectrum (b) shows a good circle shape, but the inverse of the generated disk spectrum (d) has a convex surface. Fortunately, it is not important to have a perfect flat disk. We are mainly interested in obtaining results that are rotation and scale invariant, since we need a secret function that is robust to these distortions, but we do not need a result exactly identical to $L(r)$ computed in the spatial domain.

The increase in computational efficiency is significant. Instead of 8 hours, the time for computation of r_{ref} over the same image size is reduced to approximately one minute by using frequency domain filtering. Unfortunately, computation of the binary values cannot be performed in the frequency domain. To speed up computation of the binary values, the image is over-sampled by 2 using cubic spline interpolation. Intensity values along the perimeters of the circles are computed using nearest neighbour values of the over-sampled image. This still takes about 15 minutes for a 512x512 pixels image with our implementation.

3. EXPERIMENTAL RESULTS

First, the influence of the macro-block size on watermark detection is evaluated in Section 3.1. The performance of the template hiding method that was described in Section 2 is evaluated in Section 3.2, and finally, in Section 3.3 experiments are presented to show the influence of using modulation signals with different properties.

3.1. Watermark block size

Choosing the block size is an important step in defining the ACF watermarking method, for three different reasons. First, the block size determines the number of bits that can be embedded in the watermark. The second important fact is that it determines the minimum image size for which the affine transform can be detected. If the image height and/or

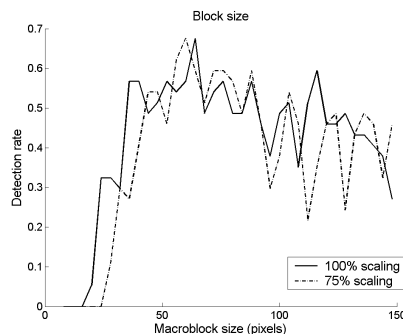


Figure 8. Influence of the size of the used watermark macro-block on the detection of the affine transform.

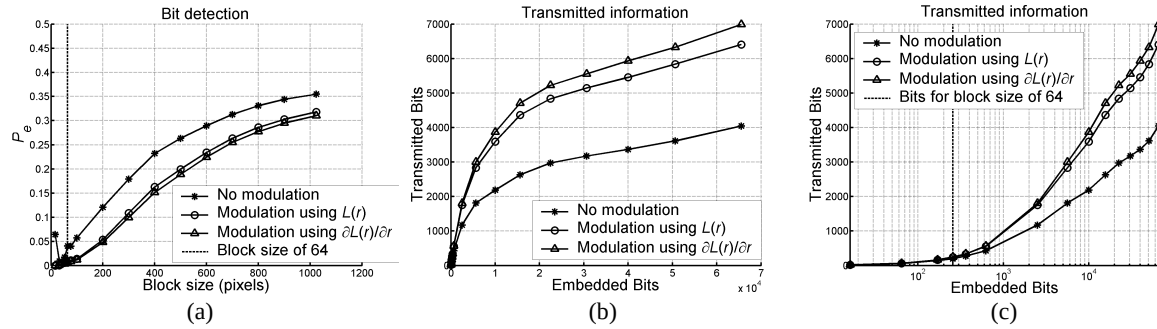


Figure 9. Influence of the size of the used watermark macro-block on the detection of the watermark bits. The bit error rate P_e is plotted in (a) as a function of the width of the block. The amount of transmitted information is shown as a function of the number of embedded bits in (b) on normal scale, and (c) on logarithmic scale, where the value of 256 bits, corresponding to a block size of 64x64 pixels, can be seen more clearly.

width is smaller than the size of one block, the periodic peaks in the autocorrelation function are absent. The last important aspect of the block size is that it will determine the grid period in the MS and ACF. This will have an influence on the detection of the grid direction and periods.

To test the influence of the macro-block size on the detection results, first the detection rate of the affine transform is tested for different block sizes. In this experiment, the watermarked image is randomly rotated. The result of detecting the affine transform, shown in Figure 8, shows a strong increase at the left side, which starts at 16 pixels when there is no scaling, but starts at 24 when the image is scaled to 75%. This is because with very small watermark blocks, the peaks in the MS are very sparse. Hence, there are less peak alignments at the correct angles of the Radon transform, reducing the detectability of the grid angles. When the image is scaled down to 75%, this effect gets even worse. With more than 50 pixels, the detection rate shows unpredictable behaviour but eventually decreases slowly. This decreasing is because although the number of peaks in the MS increases for larger block sizes, their height decreases. When the height of the peaks drops, it can drop below 'noise' level. The noise here is the spectrum of the original image that is left over after filtering and any distortion that is applied to the watermarked image. The optimal block size for detection of the affine transform in this experiment is around 64 pixels, which is the block size that will be used in the rest of the experiments.

The block size will also influence the detection of the embedded watermark bits. This is shown in Figure 9 (a). In this experiment, the bits were detected in the undistorted watermarked image for 37 different images. The bit error rate P_e increases for increasing block sizes because larger blocks will have fewer repetitions in the image. If there are fewer repetitions of a bit, its estimated value will be influenced more by noise.

However, increasing bit error rate itself does not tell anything about the performance as the number of embedded bits M also increases for increasing block sizes. The important quantity is the amount of transmitted information that is the upper bound for the amount of bits that can be transmitted when error correcting codes are used. Here, the watermarked image is actually the communication channel for the bits. Denoting with A an embedded bit value and B the corresponding detected bit value, the transmitted amount of information R can be computed by:

$$R = M \cdot I[A; B], \quad (13)$$

$$\text{with } I[A; B] = 1 - H[A; B] = 1 + \frac{1}{2} P_e \log P_e + \frac{1}{2} (1 - P_e) \log(1 - P_e). \quad (14)$$

The corresponding R for the bit detection results of Figure 9 (a) are shown in (b) and (c). It can be seen that, although the bit error rate increases for larger block size, the amount of transmitted information is the highest for the highest possible block size of 1024x1024 pixels. This block is twice the size of the image because the four quarters of the block contain the same information (see Figure 2 (b)) and only the upper left corner is embedded in the image. For the purpose of information hiding it is not efficient to repeat the same block because it adds redundancy in the available 'channel'. The channel capacity that is 'lost' by using a block of only 64x64 pixels is significant. When a watermark application requires a large watermark payload, it can be an option not to flip the watermark block. In that case, four times the

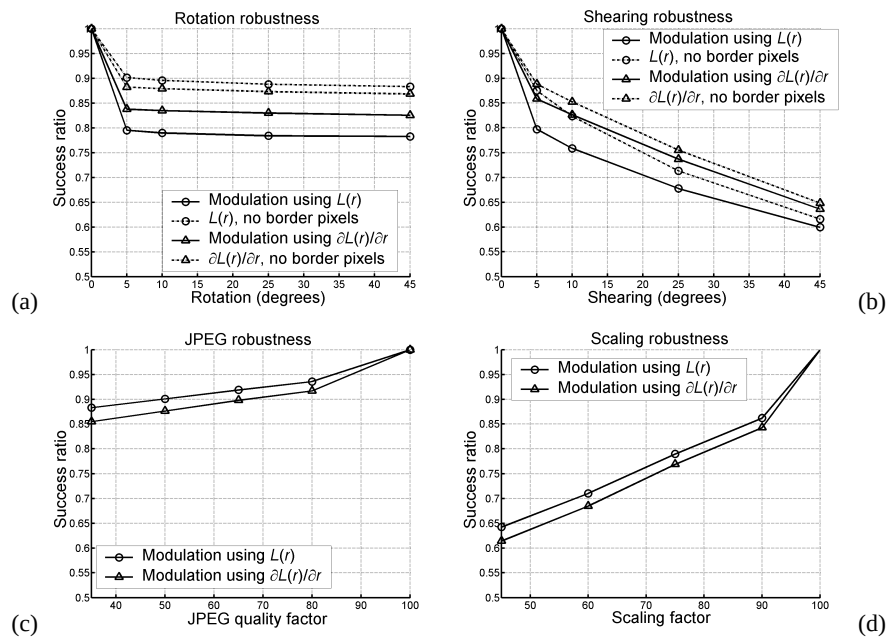


Figure 10. Robustness of the secret hiding function to rotation (a), shearing (b), JPEG compression (c) and down-scaling (d) (in percentage), denoted as the proportion of pixels that are equal for $s(i,j,k)$ of the original image and the distorted image.

amount of bits can be embedded using the same block size. The disadvantage is that the translation and orientation of the watermark block must be found by performing 2-dimensional convolution with the pilot signal 8 times when rotations over all angles and mirroring of the watermarked image can be expected.

3.2. Performance of ACF watermark template hiding

First, the robustness of the implemented hiding function to different kinds of distortions is evaluated. The effect of rotation of the image on computation of $s(i,j,k)$ is shown in Figure 10 (a). The secret modulation function $s(i,j,k)$ is computed for the original image and a rotated image (using cubic spline interpolation). The proportion of binary pixel values that are equal for the rotated modulation pattern $s(i,j,k)$ of the original image and $s(i,j,k)$ computed after image rotation is plotted as a function of the amount of rotation. For a small rotation, the result drops rapidly, but for larger rotations, the result remains fairly constant. This is because the values of the modulation signal are binary and cannot be interpolated when they are translated over sub pixel distance. Hence, the original modulation signal is rotated using nearest neighbour interpolation, which can be different from the modulation values computed from the rotated image. There is also a large difference between pixels near the border of the image and in the middle. The bounding box of the image increases due to the rotation and the empty parts are filled with zero values. This has a large influence on the binary values of all pixels for which the radii of the used circles exceed the image area. The method using $\partial L(r)/\partial r$, suffers much less from the border effect because smaller circles are used. However, it performs worse when pixels near the border are excluded. This is because interpolation errors have more influence when the mean luminance over smaller circles is determined.

The same experiment is done for shearing of the image. Here, the height of the image is kept constant, so the image area stays the same. Now the smaller scale of the method using $\partial L(r)/\partial r$ has a big advantage over the original function, as can be seen in Figure 10 (b).

For JPEG compression however, the derivative of $L(r)$ leads to much more errors, see Figure 10 (c). It is not clear if this is mainly due to the fact that noise has more influence on the mean value over small radii when computing the binary values or because the zero crossings of $\partial L(r)/\partial r$ are influenced more than zero crossings of $L(r)$.

Finally, image scaling also leads to errors, as can be seen in Figure 8 (d). This is because of the image dependency due to the low-pass filter that is used. A solution would be to define the filter scale relative to the image size, but that

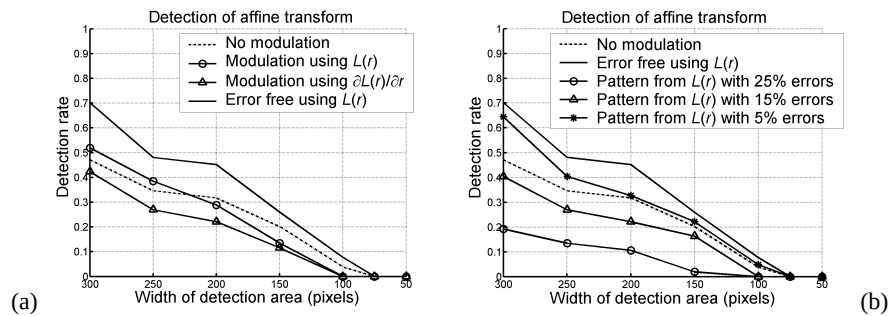


Figure 11. Results for detecting the affine transform after 45° rotation of the watermarked images. In (a) using the original hiding scheme and the one using $\partial L(r)/\partial r$. In the noise free case, the original hiding signal used for embedding the watermark was also rotated 45° instead of computing it on the rotated image. In (b) the modulation pattern from the original image was used and errors were deliberately introduced with different error rates.

would lead to more dependency on cropping of the image instead. The difference between the results for the two methods is again because of the smaller radii when using $\partial L(r)/\partial r$ that results in more influence of interpolation errors.

To see the influence of the template hiding method on the watermark detection results, the affine detection rate is measured after rotating the watermarked images over 45° . This is done for different detection area sizes to see how local the affine transform can be detected. The results are shown in Figure 11 (a). It can be seen that the original hiding scheme, using $L(r)$, gives a small improvement for detecting the affine transform on a large image area, but performs worse than the non modulated watermark for smaller image areas in this experiment. This is because for smaller areas the percentage of errors in the modulation signal has higher variance, hindering correct detection in more cases. When the exact same modulation signal is used that was used for embedding the watermark (error-free) the detection performance is always better than without using modulation. The performance for using $\partial L(r)/\partial r$ is even lower because of the higher error rate compared to $L(r)$ when no border effects occur. For comparison, in Figure 11 (b) the results are shown with different error rates of the modulation signal, artificially introduced in the rotated modulation pattern from the original image.

3.3. Spatial structure of secret modulation functions

The secret modulation function that we have used in this paper is only one example of such a function. In the future, other functions might be found that have different/better properties. To get an idea of which kind of modulation signals produce the best watermark detection results and the most robustness to errors, we also tested synthetically generated modulation patterns with artificially introduced errors. Some examples of these patterns are shown in Figure 12. These are two-dimensional versions of the random telegraph signal and are generated by thresholding a low-pass filtered zero mean Gaussian noise image at zero. The structure of the patterns is determined by the standard deviation of the Gaussian low-pass filter and is expressed as the average period between zero crossings along a straight line through the pattern.

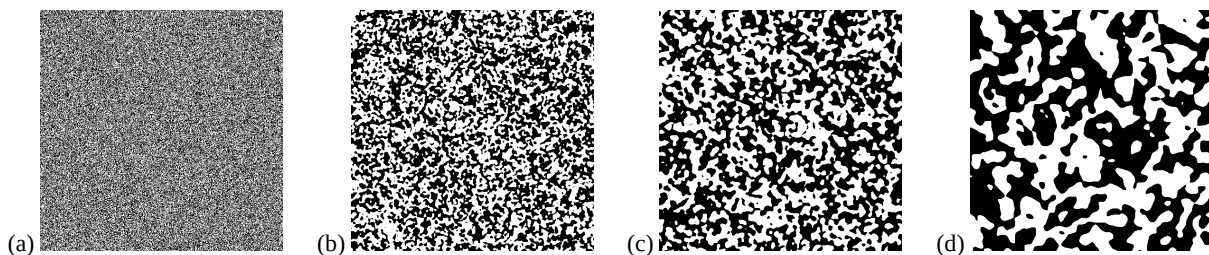


Figure 12. Pseudorandomly generated binary modulation signals with different frequencies. (a) shows complete random values, where the average period of constant binary value is approximately 2 pixels. The average period in (b), (c) and (d) is 9, 15 and 30 pixels, respectively.

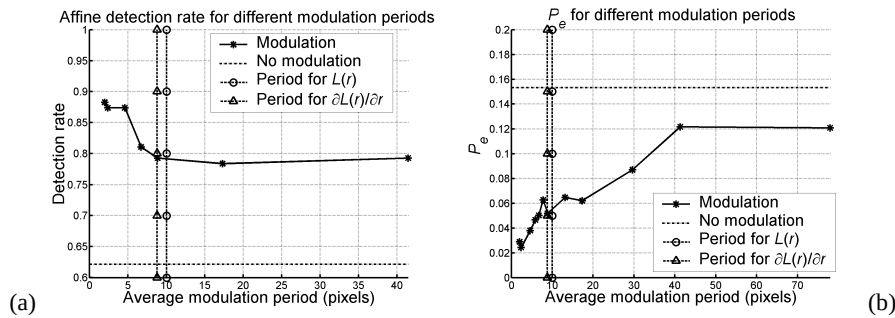


Figure 13. Influence of average modulation period on detection performance after rotating the image 45° . In (a) for detecting the affine transform, in (b) for detecting the message bits. The vertical lines indicate the periods corresponding to the existing hiding functions.

The pattern of Figure 12 (b) has an average period of 9 pixels, which is close to the average periods of the patterns of the two existing secret modulation functions, using $L(r)$ and $\partial L(r)/\partial r$ (compare to Figure 5 (c) and (f)). In Figure 13 influence of the modulation signal transition period is shown. Figure 13 (a) shows the results for detecting the affine transform after 45° rotation, and (b) for the detection of the bits after the same rotation. Both experiments show the most improvement of detection performance for modulation signals with the smallest periods. Hence, hiding functions that generate patterns with small periods would be preferred. However, hiding functions that generate these small periods may tend to be less robust to distortion. Also modulation signals with much larger periods give significant improvement over the non-modulated watermark for both the detection of the affine transform and the embedded bits. The periods belonging to the template hiding method using $L(r)$ and $\partial L(r)/\partial r$ are also shown in the graphs.

The effect of errors in the modulation signal is shown in Figure 14 for four different modulation periods: a period of 2 pixels, belonging to the random binary noise shown in Figure 12 (a), a period of 30 pixels, as shown in Figure 12 (d), and the two periods that are equal to the average periods of the hiding scheme using $L(r)$ and $\partial L(r)/\partial r$ respectively. The watermarked image is rotated 45° and the errors in the modulation signal are introduced by changing a certain percentage of binary pixel values at random locations in the rotated modulation signal. For detecting the affine transform, shown in Figure 14 (a), not much difference can be seen between the results for using modulation signals with a 30, 10 and 8.8 pixels period. Only the random noise modulation performs significantly better. This also follows from the graph of Figure 13 (a) where the largest part is nearly flat, and only using very small periods gives more performance increase. For the detection of bits, in Figure 14 (b), the differences are larger for the different modulation periods. Furthermore, here the performance stays above (P_e stays below) the detection performance of the non-modulated watermark for more modulation errors than with detection of the affine transform. For the periods of the actual hiding schemes the bit error rate stays lower for up to 30% errors in the modulation signal, while the detection rate for the affine transform already drops below the performance without modulation after 15% errors. When using the modulation signal with the smallest period more errors can be made in the modulation signal before detection performs worse than for the non-modulated watermark, for both the affine detection and detection of the watermark bits.

4. CONCLUSIONS AND FUTURE WORK

We have applied template hiding by secret modulation on ACF watermarking. This makes the watermark template robust to malicious attacks. An adjustment is proposed that improves robustness to cropping. We also proposed an efficient implementation of the hiding method using frequency domain filtering, which works much faster. We evaluated the effect on watermark detection performance of modulation of the watermark using the hiding method and also using modulation signals with different properties. The results show improvement of the watermark detection performance when the watermark is modulated. The most improvement is obtained using modulation signals with the smallest distance between zero crossings. However, hiding functions that generate these small periods may tend to be less robust to distortion. We also tested the effect of errors in the modulation signal and found that, in our implementation, successful recovery from affine distortion suffers more from modulation errors than detection of the embedded watermark bits. Using the signals with the smallest possible distance between zero crossings the watermark

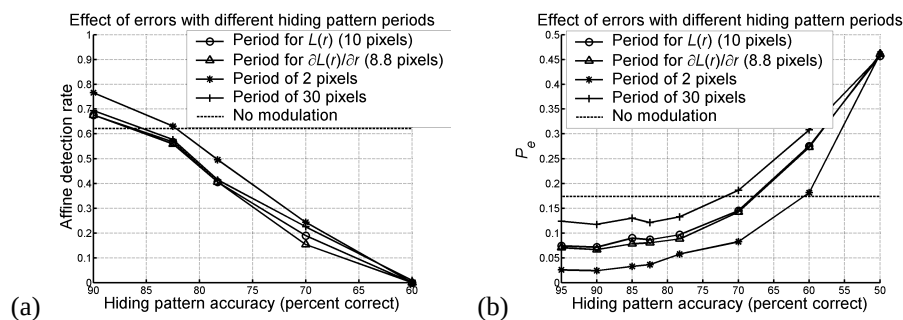


Figure 14. Influence of errors in the modulation signal on detection performance after rotating the image 45°. In (a) for detecting the affine transform, in (b) for detecting the message bits.

detector can also make the most errors in retrieving the modulation signal until the detection performance drops below the performance of the watermarking method without modulation.

Future work that remains to be investigated is to evaluate the performance of how well the periodic watermark is hidden by modulation and what the effect will be on embedding multiple watermarks in one image. It is also desirable to find other methods to generate an image dependent binary modulation signal that are faster and/or more robust to image distortion and preferably have a small period between zero crossings.

REFERENCES

1. X. Niu, M. Schmucker, C. Busch, "Video Watermarking Resisting to Rotation, Scaling, and Translation", In *Proceedings of SPIE, Security and Watermarking of Multimedia Contents IV*, Delp, Edward J.; Wong, Ping W., Vol. 4675, pp. 512-519, SPIE, San Jose, 2002
2. A van Leest, J. Haitisma, T. Kalker, "On Digital Cinema and Watermarking", In *Proceedings of SPIE, Security and Watermarking of Multimedia Contents V*, Wong, Ping Wah; Delp, Edward J., Vol. 5020, SPIE, Santa Clara, 2003
3. J. O'Ruanidh, T. Pun, "Rotation, Scale and Translation Invariant Spread Spectrum Digital Image Watermarking", *Signal Processing*, Vol. 66 (3), pp. 303-317, Elsevier, 1998
4. I. Setyawan, G. Kakes, R. Lagendijk, "Synchronization-insensitive Video Watermarking using Structured Noise Pattern", In *Proceedings of SPIE, Security and Watermarking of Multimedia Contents IV*, Delp, Edward J.; Wong, Ping W., Vol. 4675, pp. 520-529, SPIE, San Jose, 2002
5. P. Loo, N. Kingsbury, "Motion-estimation-based registration of geometrically distorted image for watermark recovery", In *Proceedings of SPIE, Security and Watermarking of Multimedia Contents III*, Wong, Ping Wah; Delp, Edward J., Vol. 4314, pp. 606-617, SPIE, San Jose, 2001
6. S. Pereira and T. Pun, "Fast robust template matching for affine resistant image watermarks," in *Proc. 3rd Int. Information Hiding Workshop*, pp. 207-218, 1999
7. S. Voloshynovskiy, F. Deguillaume, T. Pun, "Multibit Digital Watermarking Robust Against Local Nonlinear Geometrical Distortions", *Int. Conference of Image Processing 2001*, pp. 999-1002, IEEE, Thessaloniki, 2001
8. F. Deguillaume, S. Voloshynovskiy, T. Pun, "A method for the estimation and recovering from general affine transforms in digital watermarking applications", In *Proceedings of SPIE, Security and Watermarking of Multimedia Contents IV*, Delp, Edward J.; Wong, Ping W., Vol. 4675, SPIE, San Jose, 2002
9. J. Lichtenauer, I. Setyawan, T. Kalker, R. Lagendijk, "Exhaustive geometrical search and the false positive watermark detection probability", In *Proceedings of SPIE, Security and Watermarking of Multimedia Contents V*, Wong, Ping Wah; Delp, Edward J., Vol. 5020, SPIE, Santa Clara, 2003
10. M. Kutter, "Watermarking resistant to translation, rotation and scaling", *SPIE International Symposium on Voice, Video, and Data Communication*, 1998
11. D. Delannay, B. Macq, "Method for hiding synchronization marks in scale and rotation resilient watermarking schemes", In *Proceedings of SPIE, Security and Watermarking of Multimedia Contents IV*, Delp, Edward J.; Wong, Ping W., Vol. 4675, pp. 520-529, SPIE, San Jose, 2002
12. J. McClellan, "The design of two-dimensional digital filters by transformations", *Proceedings of the annual Princeton Conference on Information Sciences and Systems*, Vol. 7, p 247-251, 1973