# Image Watermarking for Tamper Detection

Jiri Fridrich

*Center for Intelligent Systems, SUNY Binghamton, Binghamton, NY 13902-6000*
*Mission Research Corporation, 1720 Randolph Rd SE, Albuquerque, NM 87501*
*fridrich@binghamton.edu*

## Abstract

*We propose an oblivious watermarking technique for tamper detection in digital images. By comparing correlation values from different portions of the image, the technique enables us to distinguish malicious changes, such as replacing / adding features from non-malicious changes resulting from common image processing operations. The technique can be implemented with small memory and computational requirements, which makes it potentially useful for hardware implementation in digital cameras. The technique works by dividing an image into blocks and watermarking each block with a transparent, robust watermark that sensitively depends on a secret key (camera's ID) and continuously on the image. The watermarking method is a frequency based spread spectrum technique. To achieve a continuous dependency on the image, we propose a special bit extraction procedure that extracts bits from each block by thresholding projections onto key-dependent random smooth patterns. Those bits are then used for initializing a PRNG and synthesizing the spread spectrum signal.*

## 1. Introduction

Powerful publicly available image processing software packages such as Adobe PhotoShop or PaintShop Pro make digital forgeries a reality. Feathered cropping enables replacing or adding features without causing detectable edges. It is also possible to carefully cut out portions of several images and combine them together while leaving barely detectable traces. Techniques such as careful analysis of the noise component of different image segments, comparing histograms of disjoint image blocks, or searching for discontinuities could probably reveal some cases of tampering, but a capable attacker with enough expertise can always avoid such traps and come up with an almost perfect forgery given enough time and resources. This is one of the reasons why digital imagery is not acceptable as evidence in establishing the chain of custody in the court of law. There are other instances, of mostly military character where image integrity is of paramount importance. Digital watermarking can be used as a means for efficient tamper detection. One could mark small blocks of an image with watermarks that depend on a secret ID of that particular digital camera and later check the presence of those watermarks. The "fragility" of the watermark against various image distortions determines our ability to measure the extent of tampering.

One of the first techniques used for detection of image tampering was based on inserting check-sums into the least significant bit (LSB) of image data. Walton [1] proposes a technique that uses a key-dependent pseudo-random walk on the image. The check-sum is built from the 7 most significant bits and is inserted in the LSB of selected pixels. To prevent tampering based on exchanging groups of pixels with the same check-sum, the check-sum is "walk-dependent". Although check-sums can provide a very high probability of tamper detection, they cannot distinguish between an innocent adjustment of brightness and replacing a person's face. Increasing the gray scales of all pixels by one would indicate a large extent of tampering, even though the image content has been unchanged for all practical purposes. Van Schyndel et al. [2] modify the LSB of pixels by adding extended m-sequences to rows of pixels. The phase of the sequence carries the watermark information. A simple cross-correlation is used to test for the presence of the watermark. As with any LSB technique, this method will provide a low degree of security and will not be robust to image operations with low-pass character. Wolfgang and Delp [3] extended van Schyndel's work and improved the localization properties and robustness. They use m-sequences of –1's and 1's arranged into 8×8 blocks and add them to corresponding image blocks. Their technique is moderately robust with respect to linear and nonlinear filtering and small noise adding. Since the watermark is inserted in the LSB plane, it can be easily removed. Zhu et al. [4] propose techniques based on spatial and frequency masking. Their watermark is guaranteed to be perceptually invisible, yet it can detect errors up to one half of the maximal allowable change in each pixel or frequency bin depending on whether spatial or frequency masking is used. The image is divided into blocks and in each block a secret random signature is modulated by the

masking values of that block. The error estimate is fairly accurate for small distortions. It is unclear, however, if this technique would provide any useful information for images that have been distorted by more than a perceptually invisible amount. Even though the image has been visibly distorted, we might want to argue that the image content is essentially the same and no large malicious changes occurred. This could be done using a robust watermarking scheme applied to larger blocks. The watermark in method [4] depends on the image in a weak manner. The secret signature does not depend on the image – it is modulated by the masking values of each block. But those masking values are available to anybody to compute. Marking a large number of images with one secret key would be obviously insecure. Such a technique would not be suitable for marking images in digital cameras.

In this paper, we describe a technique that uses a robust watermark in larger blocks (i.e., 64×64 pixels). To prevent unauthorized removal or intentional distortion, the watermark must depend on a secret key $S$ (camera's ID), block number $B$, and on the content of the block. The content of each block is represented with $M$ bits extracted from the block by projecting it on a set of random, smooth patterns and thresholding the result. This extraction process gives similar $M$-tuples for similar blocks enabling thus a successful synthesis of the spread spectrum signal from the watermarked / tampered image. The spread spectrum signal for each block is generated by adding $M$ pseudo-random sequences uniformly distributed in $[-1,1]$. Each sequence depends on the secret key, block number, and the bit extracted from the block. If $k$ out of $M$ bits are extracted incorrectly due to image distortion, the spread spectrum signal will still have large correlation with the image as long as $k << M$.

The spread spectrum signal is rescaled, made DC-free, and added to the middle third of DCT coefficients for each block. The detection proceeds by blocks by recovering $M$ bits from each block, generating the spread spectrum signal, and correlating it with the middle third of DCT coefficients of that block.

If watermarks are present in all blocks with high probability, one can be fairly confident that the image has not been tampered with in any significant manner (such as adding or removing features). If the watermark correlation is lower uniformly over all image blocks, one can deduce that some image processing operation was most likely applied. Based on the image content and the watermark strength in each block one can further attempt to classify which image operation was applied (e.g., low-pass filter, high-pass filter, gamma correction, noise adding, etc.). If one or more blocks show very low evidence for watermark presence while other blocks exhibit values well above the threshold, one can estimate the probability of tampering and, hopefully, with a high probability decide whether or not the image has been tampered with.

In Section 2 we give details of a new watermarking technique for tamper detection and present some experimental results. Future directions, possible improvements, and implementation issues are discussed in Section 3.

## 2. Description of the technique

Watermarking for tamper detection that would be implemented in digital cameras has its own specifics. In one possible scenario, a special tamper-proof watermarking chip inside a digital camera will watermark the image data before it is stored on camera's memory media (e.g., hard disk, flash card, or tape). We note that in this particular case, the original unwatermarked image will never be produced. Therefore, the watermarking scheme must be oblivious. Clearly, it is important that the watermark be perceptually invisible so that the image quality is preserved. It is equally important that the technique has low computational complexity and low memory requirements. The watermark must depend on the image and on a secret camera ID. It should survive common image processing operations, such as contrast/brightness adjustment, blurring, sharpening, noise adding, and lossy compression. However, there is a conflict between robustness and the size of the block. While is desirable to protect as small portions of the image as possible, smaller image blocks inevitably decrease the robustness. As a trade-off between these conflicting requirements, we opted for block sizes of 64×64 pixels. In our choice, we were lead by the fact that a human face scaled to a 32×32 block is of such a low resolution that an identification becomes impossible.

The technique proposed in this paper starts with dividing the image into small blocks of 64×64 pixels. Each block is watermarked using a frequency based spread spectrum technique similar to the one proposed by Ó Ruanaidh [5]. Denoting the $i$-th block by $B_i$, we carry out the following three steps for each block:

**Step 1 (Extracting $M$ image content bits).** Due to security reasons, the watermark pattern must depend on the block. The goal is to design a robust procedure for extracting $M$ (~30) bits from each block. On the one hand, it is important to have uncorrelated $M$-tuples for different blocks and different images, on the other hand, the $M$-tuples should be almost identical for all similar looking blocks. Using a PRNG seeded with camera's ID, we generate $M$ random black and white patterns $P_i$ of the same size as the blocks, smooth them using a low-pass filter, and make them DC-free. Then, we calculate the projections of those patterns on the image block. We experimented with blocks of many different images to find

out the distribution of those projections. The distribution appears to be Gaussian (see Figure 1).

If the projection on a particular pattern is large, it is unlikely that small image distortion will change it to a small value and vice versa. Therefore, it makes sense to extract one bit $b_i$ from each projection by thresholding its absolute value with a suitable threshold $T_p$,

$$b_i = 1 \text{ if } |P_i \cdot B_i| > T_p$$
$$b_i = 0 \text{ otherwise.}$$

The threshold $T_p$ was chosen so that approximately half of the extracted bits are ones and the other half zeros. This way, the extracted $M$-tuples will have the highest information content. In our experiments, we took $T_p$ = 2500. We tested the bit extraction procedure for 64×64 blocks of the test image "Lenna". Out of 50 bits, we were

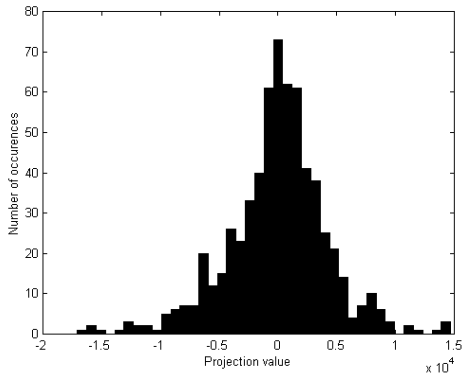


**Figure 1 Distribution of projections onto random smooth patterns.**

able to recover at least 47 bits correctly (some blocks had more correct bits) after applying a blurring operation (as in PaintShop Pro) four times. Adjusting brightness by ±25%, which resulted in unacceptably light or dark images, lead to at least 45 and 44 correct bits, respectively. Adding white Gaussian noise with standard deviation of 36 gray levels resulted in at least 46 correct bits. Other common image operations, such as histogram equalization, sharpening, decreasing color depth, and JPEG compression with quality factors as low as 10%, produced similar results. Using error correction, one can reliably extract 30 bits from each block. The ratio of ones and zeros in those extracted 50-tuples was close to ½. There are two reasons why we used smoothened random patterns rather than white noise patterns. Smooth patterns are less sensitive to synchronization, which might be important if the image has been cropped or resized and a search must be performed. Second, our experiments indicate that bit extraction based on smooth patterns is more robust with respect to image distortions.

**Step 2 (Generating the spread spectrum signal).** Since the watermarking technique from Step 3 modulates the middle third of DCT coefficients ($D$ coefficients) using a spread spectrum signal, we generated $M$ pseudo-random sequences of length $D$ uniformly distributed in [0,1], added them together, and adjusted to a predefined standard deviation and zero mean. To generate the $j$-th sequence in block $B_i$, $1 \le j \le M$, with the $j$-th extracted bit $b_j$ we seeded a PRNG with a concatenation of camera's ID $S$, $i$, $j$, and $b_j$. It is important that sequences from different image blocks and for different extracted bits $b_j$ are uncorrelated. This is the reason why the seed contains the block number $i$ and the sequence number $j$ explicitly.

In our implementation, we actually used the approach described in [5] and hid a sequence of $M$ symbols each symbol consisting of $r$ bits in the spread spectrum signal. To hide $M$ $r$-bit symbols, we generate $M$ pseudo-random sequences of length $D$, each sequence chosen randomly as a segment of $D$ numbers out of $D+r$ randomly generated numbers. The spread spectrum signal is then obtained as sum of those signals. To detect which symbol is hidden, one simply calculates cross-correlation of the recovered $D$ DCTs with shifted versions of the generated $D+r$ sequences. For details, see [5]. In our experiments, we embedded one fixed symbol $M$-times thus sacrificing capacity of the watermark for robustness.

**Step 3 (Inserting the watermark).** We calculate the DCT of each block and modulate the middle 30% of DCT coefficients by adding the spread spectrum signal. The amplitude of the added signal can be adjusted to achieve balance between watermark visibility and robustness. We set the amplitude equal to 13 (we used the symmetric form of DCT). Using the linearized spatial masking model of Girod [6] without the temporal aspect, the watermark was visible for 0.17% of all pixels.

The detection of the watermark proceeds by blocks. For each block, $M$ bits are extracted and the block is DCT transformed. Then, the spread spectrum signal is synthesized using the camera ID and the PRNG. Total $M$ symbols are extracted from each block by choosing the symbols with the largest correlation. For each block, we add the number of correctly recovered symbols and calculate the probability of obtaining that many correct symbols. With $M$ $r$-bit symbols, the probability $P(k,M)$ of getting at least $k$ correct symbols out of $M$ symbols is

$$\binom{M}{k} 2^{-rk}.$$

The threshold for watermark presence, or evidence that the block has not been tampered with, should be based on this probability. For example, $P(5,10) = 2.3\times10^{-7}$, which means that the probability of obtaining at least five correct symbols out of 10 is less than 1:4,000,000. Replacing a block or detecting the watermark with a wrong key leads

to larger values of $P$. Figure 2 shows the maximum of $P(k,M)$ taken over all 16 blocks in a 256×256 image for 1000 randomly generated secret keys.

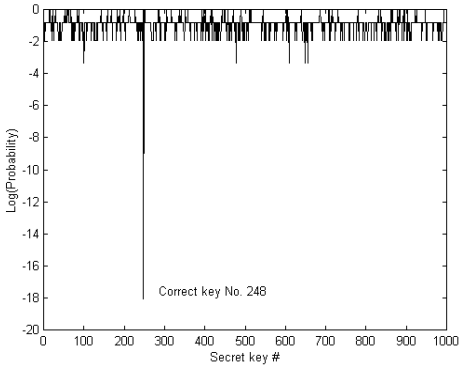


**Figure 2 Results of testing watermark presence using 1000 random keys.**

Due to the nature of the watermarking scheme, the watermark is fairly robust with respect to contrast/brightness adjustment, histogram equalization, noise adding, and sharpening. It also survived in all blocks fairly well for one and two consecutive blurring operations (in PaitShop Pro). The robustness with respect to JPEG coding was less satisfactory. At 50% quality JPEG compression, some blocks indicated a weak watermark presence even though the watermark survived in most of the blocks. Detailed study of the robustness of this technique with respect to image distortion will be described in a forthcoming paper [7]. We are currently studying alternative watermarking schemes, such as the scheme proposed by Swanson [8], and its suitability for tamperproofing.

To further test the scheme, we have cropped a rectangular portion of the unwatermarked image (see Figure 3) with feather width 11 and pasted it into the corresponding watermarked image. There was absolutely no visible indication of tampering in the tampered watermarked image. We applied a detection function with the correct camera ID. The result is shown in Figure 4. The graph nicely reflects the fact that a large portion of block No. 10 and 11 has been replaced – the probability $P$ of tampering is very close to one. Non-tampered blocks have the probability of tamper equal to $10^{-18}$. The blocks No. 6 and 7 have been replaced only partially which is indicated by higher probability values, which are nevertheless still very small. This indicates that the watermark is also fairly robust to cropping. The last two tampered blocks No. 14 and 15 exhibit only a slight increase in the probability of tampering.

It is evident that the watermark robustness directly influences the sensitivity of the tamper detection procedure. On the one hand, watermarks that are too sensitive to small distortions will be too easy to remove by common image processing operations. This will diminish our ability to discern between malicious attacks and innocent image adjustments. On the other hand, a robust watermark may not be able to detect small malicious changes in portions of the block (due to robustness to cropping). Therefore, it makes sense to actually combine some form of LSB check-sum encoding [3] with our technique. This will enable us to detect a wider spectrum of possible image modifications. Check-sums will be useful for detection and localization of small, localized changes, while the robust watermark may help tremendously if the image has also been processed.

## 3. Improvements and future directions

For practical implementation, if the random smooth patterns are not stored but generated each time a picture is taken, the total memory requirements are approximately determined by the number of pixels in two blocks plus the length of the spread spectrum signal. This gives us roughly 9.3kB. Calculating the patterns for each picture is however not necessary and the watermarking process can be sped up by precalculating the patterns and storing them inside the camera. If $M$ = 30 patterns is used, we will need storage for $30\times64^2$ bytes = 123kB.

Embedding an additional calibration signal for detection of rotation and scaling as in [9] will improve the efficiency of the process of tamper detection significantly.

To improve the robustness with respect to low-pass filtering and low-quality JPEG coding, the watermark could be combined with a low-frequency watermark [10].

In our experiments, we have noticed that the detection of watermark in each block highly depends on the block content. Some image deformations leave the watermark practically unchanged in certain blocks, while other blocks indicate that the watermark is present weakly. Usually, blocks with features or textured blocks more easily retain the watermark than blocks with relatively flat content. We may attempt to categorize different image blocks based on their content and estimate the watermark's sensitivity with respect to specific image distortions. This may help in distinguishing between the loss of correlation due to cropping / replacement and applying some image processing operation.

The watermark strength should be adapted according to the block content. Perceptual models of the human visual system, and frequency and spatial masking will likely produce more reliable watermark. However, for small, relatively flat blocks there may not be much that could be done because the robustness of a watermark in such areas will always be low no matter which watermarking technique is used.

## 4. Acknowlegements

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under a Phase II SBIR grant number F30602-98-C-0049. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U. S. Government.

## 5. References

[1] S. Walton, "Information Authentication for a Slippery New Age", *Dr. Dobbs Journal*, vol. 20, no. 4, pp. 18–26, Apr 1995.

[2] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A Digital Watermark", *Proc. of the IEEE Int. Conf. on Image Processing*, vol. 2, pp. 86–90, Austin, Texas, Nov 1994.

[3] R. B. Wolfgang and E. J. Delp, "A Watermark for Digital Images", *Proc. IEEE Int. Conf. on Image Processing*, vol. 3, pp. 219–222, 1996.

[4] B. Zhu, M. D. Swanson, and A. Tewfik, "Transparent Robust Authentication and Distortion Measurement Technique for Images", preprint, 1997.

[5] J. J. K. Ó Ruanaidh and T. Pun, "Rotation, Scale and Translation Invariant Digital Image Watermarking", *Proc. of the ICIP*, vol. 1, pp. 536–539, Santa Barbara, California, Oct 1997.

[6] B. Girod, "The Information Theoretical Significance of Spatial and Temporal Masking in Video Signals", *Proc. of the SPIE Human Vision, Visual Processing, and Digital Display*, vol. 1077, pp. 178–187, 1989.

[7] J. Fridrich, "Methods for Detecting Changes in Digital Images", *Proc. of The 6th IEEE International Workshop on Intelligent Signal Processing and Communication Systems* (*ISPACS'98*), Melbourne, Australia, 4–6 November 1998.

[8] M. Swanson, B. Zhu, and A. H. Tewfik, "Data Hiding for Video-in-video", *Proc. ICIP '97*, vol. II, pp. 676–679, 1997.

[9] A. Herrigel, J. O Ruanaidh, H. Petersen, S. Pereira, T. Pun, "Secure Copyright Protection Techniques for Digital Images," *Proc. 2nd Int. Information Hiding Workshop*, Portland, Oregon, Apr 1998.

[10] J. Fridrich, "Combining Low-frequency and Spread Spectrum Watermarking", *Proc. SPIE Int. Symposium on Optical Science, Engineering, and Instrumentation*, San Diego, July 19–24, 1998.

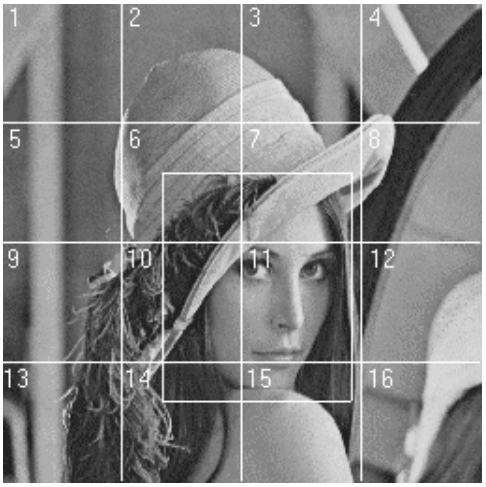


**Figure 3 Image of "Lenna" divided into blocks of 64x64 pixels. The rectangular region has been replaced with corresponding part from the original unwatermarked image.**

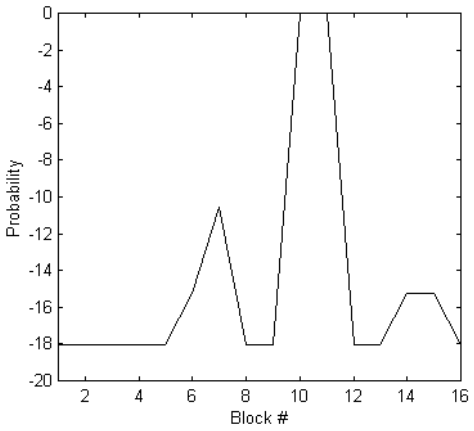


**Figure 4 Detection of malicious changes.**