METHODS FOR DETECTING CHANGES IN DIGITAL IMAGES

Jiri Fridrich
Center for Intelligent Systems, SUNY Binghamton, Binghamton, NY 13902-6000
Ph/Fx: (607) 777-2577
fridrich@binghamton.edu
Mission Research Corporation, 1720 Randolph Rd SE, Albuquerque, NM 87501

ABSTRACT

The goal of this paper is to design a watermarking scheme capable of distinguishing visible but non-malicious changes due to common image processing operations from malicious changes, such as feature adding/replacement. We start with an overview of watermarking techniques for detection of tampering in digital images and discuss their limitations. A new technique that inserts robust watermarks into small disjoint blocks is proposed. The technique can be implemented with small memory and computational requirements, which makes it potentially useful for hardware implementation in digital cameras. This paper further extends our previous effort [1].

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 images typically contain a lot of redundant information due to large spatial correlations. It is possible to introduce a large MSR error but still be able to identify important features in the image. A good method for detection of tampering should be able to distinguish small, unimportant changes due to common image processing operations from malicious changes, such a erasing features, adding new features, etc. The newly emerged field of information hiding provides new, versatile, and powerful tools for detection of tampering in digital images.

Digital watermarking can be used as a means for efficient tamper detection in the following way. 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.

1.1 Embedding Check-Sums in LSB

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. Images taken with CCD elements or scanned on a scanner always contain a noise component. Hiding check-sum bits in the LSB will not produce visible changes. Walton [2] proposes a technique that uses a key-dependent pseudo-random walk on the image. The check-sum is obtained by summing the numbers determined by the 7 most significant bits and taking a remainder operation with a large integer N. The probability that two groups of pixels will have the same check-sum is $1/N$. The check-sum is inserted in a binary form in the LSB of selected pixels. This could be repeated for many disjoint random walks or for one random walk that goes through all pixels. To prevent tampering based on exchanging groups of pixels with the same check-sum, the check-sum can be made "walk-dependent". The method is very fast and on average modifies only half of the pixels by one gray level. 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.

1.2 Embedding M-sequences

Van Schyndel et al. [3] modify the LSB of pixels by adding extended m-sequences to rows of pixels.

The sequences are generated with a linear feedback shift register with n-stages with periods as high as 2^n. M-sequences have known desirable autocorrelation and randomness properties. For an $N \times N$ image, a sequence of length N is randomly shifted and added to the image rows. The phase of the sequence carries the watermark information. A simple cross-correlation is used to test for the presence of the watermark. This technique is robust to small amount of noise and can accommodate more than one watermark because different segments of m-sequences are uncorrelated. The watermark can, however, be easily removed or replaced by manipulating the LSB. In addition to that, the method does not have good localization properties. Wolfgang and Delp [4] extended van Schyndel's work and improved the localization properties and robustness. They use bipolar 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 last two LSBs, again, it can be easily removed.

1.3 Distortion Measure Based on Perceptual Watermarking

Zhu et al. [5] propose two 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 [6] or frequency [7] masking is used. The image is divided into blocks and in each block a secret random signature (a pseudo-random sequence uniformly distributed in [0,1]) is multiplied by the masking values of that block. The resulting signal depends on the image block and is added to the original block quantized using the same masking values. The changes are thus always less than or equal to the maximal allowable change and do not introduce visible artifacts. Errors smaller than one half of the maximal allowable change are readily detected by this scheme. The error estimates are 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 this method [5] 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.

1.4 A New Block-Watermarking Technique

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 depends 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 results. This extraction process gives similar M-tuples for similar blocks enabling thus a successful synthesis of a 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.

2. NEW WATERMARKING 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 method must be oblivious and be able to detect changes without accessing the original image. Clearly, it is important that the watermark be perceptually invisible so that the image quality is preserved. Because we envision that the technique will be implemented in hardware in a digital camera, the technique must have 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 it 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.

2.1 Watermark Insertion

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 [8]. 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. We extract M (~30) bits from each block and build a spread spectrum noise-like signal from this M-tuple. Since we need to be able to extract the watermark from distorted images, we need a procedure that would give us the same or similar M-tuple for all similar looking blocks. We seeded a PRNG with a secret camera's ID and generated M random black and white patterns P_i of the same size as the blocks. The blocks were then smoothened using a low-pass filter, and made DC-free. 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 operation $A \cdot B$ for two matrices A and B is defined as sum of an element-wise product of both matrices. The threshold T_p was chosen so that approximately half of the extracted blocks are ones and the other half zeros. This way, the extracted M-tuples will have the highest information content. In our experiments, we took $T_p = 2500$. For more details on this extraction technique, see [1].

Step 2 (Generating the spread spectrum signal). In this step, we generate the spread-spectrum signal that will be added to the middle third of the DCT of the corresponding block ($D = M \times N/3$ coefficients). For each block B_i, we generate M pseudo-random sequences of length D uniformly distributed in [0,1], add them together, and adjust to a predefined standard deviation and a 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 seed a PRNG with a concatenation of camera's ID, S, block number i, bit number j, and extracted bit b_j.

In our implementation, we actually used the approach described in [8] 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 a 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 [8]. 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 a 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.

2.2 Watermark Detection

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 a high value of P. We tested the value of $P(k,M)$ for a 256×256 image for 1000 randomly generated secret keys. Detection with the wrong key or replacing a block lead to large values of P (e.g., in the interval $[10^{-4},1]$). Untampered blocks had values of P close to 10^{-18}.

3. ROBUSTNESS TO COMMON IMAGE PROCESSING OPERATIONS

As can be expected for a spread-spectrum technique, the watermark is fairly robust with respect to brightness/contrast adjustment, noise adding, histogram manipulation, and cropping. All image deformations were done using PaintShop Pro 4.12. Figures 1 and 2 indicate that even after 50% adjustment of contrast (as in PaintShop Pro 4.12) we could verify the integrity of the image. The watermarks in all blocks have survived for ±25% adjustment in brightness (Figure 3). Figure 4 demonstrates the robustness with respect to noise adding. Even though white Gaussian noise with standard deviation of 21 gray levels produced quite unacceptable image distortion, one could still establish the presence of watermarks with very high probability for most of the blocks. Although the watermark is spanned by middle frequencies, it has survived repeated low-pass filtering. Figure 5 shows the results for one and two consecutive applications of the blurring filter. High-pass filtering (Sharpen More in PaintShop) as well as histogram equalization did very little damage to the watermark (see Figure 6). The watermark survived JPEG compression with moderate quality factor (up to 55% quality). Lower quality factors produced some blocks for which the watermark presence could not be reliably established. Robustness to JPEG compression is obviously very important because many digital cameras store the information in a compressed form to save the storage space. We could improve the robustness by adding another low-frequency watermark that would not interfere with the spread-spectrum watermark. Another possibility would be to watermark directly the JPEG stream instead of the raw image data. We are currently testing other watermarking techniques [9] and their suitability for tamper proofing.

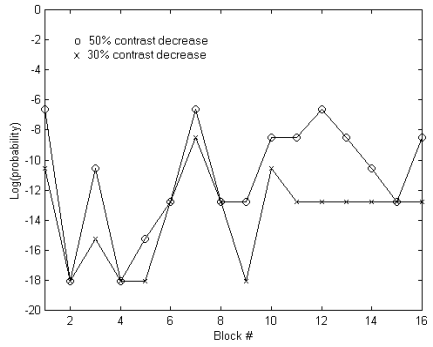

Figure 1 Robustness to contrast decrease.

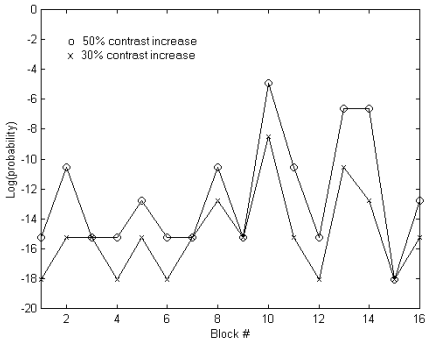

Figure 2 Robustness to contrast increase.

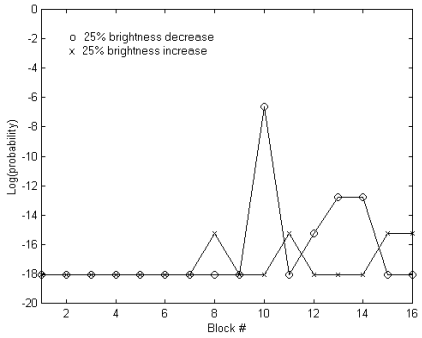

Figure 3 Robustness to brightness adjustment.

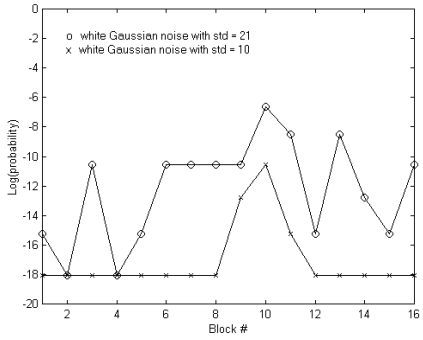

Figure 4 Robustness to white Gaussian noise.

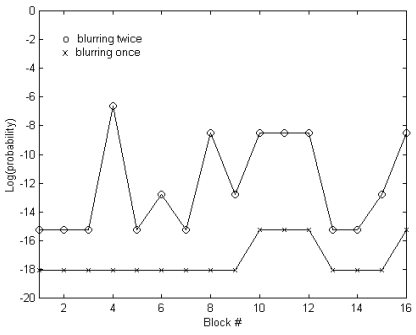

Figure 5 Robustness to repeated blurring.

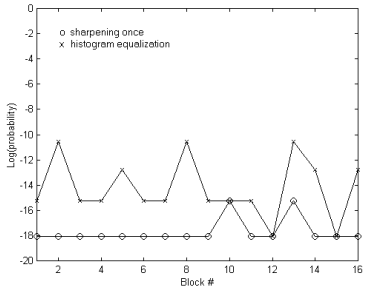

Figure 6 Robustness to sharpening and histogram equalization.

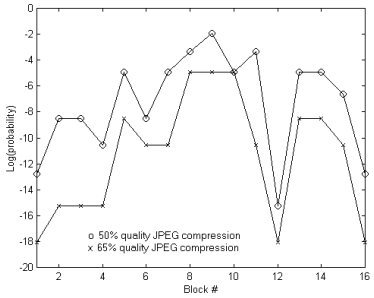

Figure 7 Robustness to JPEG compression.

3. SUMMARY AND FUTURE DIRECTIONS

In this paper, we overviewed current techniques for tamper detection in digital images. We also proposed and tested a new technique based on watermarking blocks of 64×64 pixels with a transparent robust watermark pattern. The pattern is generated by modulating the middle frequencies of the blocks' DCT with a spread spectrum noise-like signal. The signal is produced from a PRNG seeded with camera's ID, block number, and bits extracted from the block. The watermark is embedded in a robust manner and cannot be removed without introducing visible distortions into the image. It enables us to distinguish visible non-malicious changes due to common image processing operations from malicious modifications, such as replacing or adding features.

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] J. Fridrich, "Image Watermarking for Tamper Detection", *Proc. ICIP '98*, Chicago, Oct 1998.

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

[3] 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.

[4] 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.

[5] B. Zhu, M. D. Swanson, and A. Tewfik, "Transparent Robust Authentication and Distortion Measurement Technique for Images", preprint, 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] G. E. Legge and J. M. Foley, "Contrast Masking in Human Vision", *J. Opt. Soc. Am.*, **70**(12), pp. 1458–1471, 1980.

[8] 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.

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