NRL Release Number 03-1221.1-2429

THE BINARY PHASE ONLY FILTER AS AN IMAGE WATERMARK

Center for High Assurance Computer Systems Tech Memo 5540—38TM
released 12/4/03, (available to public) posted 1/12/04

Farid Ahmed[1] *& Ira S. Moskowitz*[2]

[1]Dept of EE & CS
The Catholic University of America
Washington, DC 20064
ahmed@cua.edu

[2]Center for High Assurance Computer Systems
Naval Research Laboratory---5540
Washington, DC 20375
moskowitz@itd.nrl.navy.mil

ABSTRACT

We describe our new method for watermarking digital images. Our work is motivated by the study of phase only filters in Fourier optics. In this paper we concentrate on greyscale images, even though our method works for color also. We take the discrete Fourier transform of an image and determine a signature based upon a binary phase-only filter (BPOF). We replace certain frequency magnitudes with this BPOF. This serves as the basis for our watermark. We may also insert additional side information and our method prevents spoofing of the watermark. Our method survives JPEG compression so that the watermark survives to pass various correlation tests. Our watermarking scheme is used for authentication purposes.

1. INTRODUCTION

Watermarking of digital greyscale images (*image* for short) is part of the larger area of information hiding. We concentrate on a method of watermarking that can be used to prove the authenticity of an image. To be precise, if Bob receives an image from sender Alice and our watermark detector detects a watermark, then Bob can be assured that the image was sent by Alice, and that at worst, the image has only been trivially tampered with. The watermark detector does not need Alice's original image, thus our method is *blind*. Our method allows some corruption of the image in order to survive JPEG compression, thus our method is *semi-fragile*. Our method has an embedder, used by the sender, and a detector, used by the receiver. We use a binary phase-only filter (BPOF), which is an inherent characteristic of an image, as the basis for our watermark.

One starts with the spatial realization of an image---the luminance values of the pixels (*bitmap*). Then one applies the 2-dimensional discrete Fourier transform (DFT) to the bitmap. The transmitter wishes to hide the BPOF in the image, thus forming the watermarked image. This hiding is done by changing, in a specific manner, the magnitude of the Fourier coefficients, based upon the BPOF. This change is done in a judicious manner so that the image is not visually degraded, while at the same time the changes are large enough so as not to be lost in the "noise" of compression. Now the 2-dimensional inverse discrete Fourier transform (IDFT) is applied and one arrives at a modified bitmap in the spatial domain. This is the watermarked image.

The watermarked image is saved either without compression, such as a TIFF, or with compression, such as a JPEG. This watermarked file is emailed, posted, put on a web page, etc. by the sender to the receiver(s). This received file is converted into a bitmap and transformed into the frequency domain by the DFT. Thus the BPOF is extracted from the watermarked image. The receiver runs the watermark detector which determines the hidden information from the Fourier magnitudes. This extracted information is compared to the BPOF. If there is a match, then the image is considered to be authentic.

It is possible for someone to spoof a simple embed and extract approach such as described above, especially if they are aware of the algorithm (e.g., Kerckhoffs' principle [2]). Therefore, both the watermark embedder and watermark extractor have cryptographic plug and play components to them. Therefore, our method cannot be spoofed. Our method can be attacked so that the watermark is removed and hence a false negative can be achieved, but our method will not give false positives.

The matching of the extracted information to the BPOF is accomplished by using various correlation metrics. Our algorithm is implemented as Matlab code. .

2. BPOF AS AN IMAGE SIGNATURE

We use the binary phase-only filter (BPOF) [5,6] as a signature of the original image. Previous work in image analysis has demonstrated that Fourier phase is more important in image reconstruction than Fourier magnitude [7]. Following this, Horner and Gianino proposed a computer simulation of optical correlation based on a phase-only filter (POF) [5]. The POF, and a number of its later variants, have demonstrated significant improvement in correlation performance compared to the classical matched filter when used in pattern recognition applications [4,6]. The improvement is largely attributed to emphasizing the high-frequency components of the filter. Soon after, primarily motivated by recording the filter in the binary spatial light modulator, the binary phase-only filter (BPOF) was proposed independently by two research groups [e.g. 5]. A number of binarization techniques were also proposed subsequently. While it maintained most of the correlation performance of the POF, BPOF correlation also yields computational advantages. On another perspective, variants of phase-only filters were also shown to good features for pattern recognition applications [1]. Because of the above reasons we propose using the BPOF as a signature of the image, which will be hidden as a watermark in the original image. In the detection stage of the authentication application, the hidden signature will be correlated with the extracted signature. There is another interesting aspect of selecting BPOF as a signature. The phase quantization in BPOF has built in tolerance to some minor changes to the image which is an attractive feature of semi-fragile image authentication.

3. WATERMARKING PROCESS

Consider a bitmap image *h(m,n),* where *m,n* are the spatial indices (pixel locations) for an MxN image. The DFT of *h(m,n)* is written as *H(u,v)* where *u,v* represent the frequency coordinates. (Note we use the FFT implementation of the discrete Fourier transform in Matlab which normalizes terms in the inverse transform.)

$$(1) \qquad H(u,v) = | H(u,v) | \exp(j\phi(u,v)) .$$

|H(u,v)| is the *magnitude* of coefficient *H(u,v)* and $\phi\,(u,v)$ is the *phase*, where *j* is the principal valued square root of -1. The magnitudes span a range of values from 0 to MAX, and the phases are in the interval (-π,π].

The *phase-only filter* (POF) [5] is obtained by setting the magnitude of all frequencies to 1, and by taking the complex conjugate (we use the conjugate to facilitate the correlation test) of the phase portion, thus, the POF is given by the unit complex number

$$(2) \qquad H_{POF}(u,v) = \exp(-j\phi(u,v)) .$$

We now further filter the POF by thresholding it and thus obtaining a BPOF [6]. We use the BPOF given below

$$(3) \quad H_{BPOF}(u,v) = \begin{cases} 1, \cos(\phi) \geq 0 \\ \text{-1, otherwise} \end{cases}$$

Thus, to each frequency we assign a bit. This bi-polar representation of the BPOF is used for the correlation tests, however, when we embed the watermark into the Fourier magnitudes as discussed below the BPOF takes on the value 1 or 0 (corresponding to -1). We let $\mathbf{S}$ represent the {0,1} BPOF. (Correlation tests perform better on {-1,1} than they do on {0,1}.)

The magnitude after integer rounding *|H(u,v)|* is stored (in Matlab) as an N-digit binary number depending on what MAX is. We concentrate our efforts on modifying the various bit planes that comprise the magnitude values. Our method is not limited to one specific bit plane, but for the sake of simplicity we will use the 13th bit plane (experiments have shown this to be a good value for 512x512 images) in this paper. In general the mid-level bit plane seems to be a good compromise between watermark survivability and minimal visual effect upon the original image. Keep in mind that the BPOF inherits a symmetry from the behavior of the Fourier frequencies
(*H(M-u,N-v)* = *H*(u,v)*, where * is complex conjugation). We encrypt $\mathbf{S}$ via symmetric encryption (no frequency dependency) with key k to form $E_k(\mathbf{S})$ (the encryption must respect the symmetry of the BPOF). This is what we hide. Again, the encryption is needed to prevent spoofing. However, the encryption has the additional positive effect of minimizing any biases that would occur in the following procedure if we simply used $\mathbf{S}$.

EMBEDDER PROCESS: Now we replace the selected bit plane with $E_k(\mathbf{S})$. Since this only changes magnitudes we have not changed any frequency phases. Thus we have formed a modified *H*(*u,v*), we now take this modified *H*(*u,v*) and apply the IDFT to it. This takes us back to the spatial domain. We integer round and clip at 0 and clamp at 255 these values. Thus we are left with a valid spatial image W---the watermarked image. We may store this image in compressed form such as a TIFF or in compressed form as a JPEG, etc.

DETECTOR PROCESS: Now the recipient of W must reverse the process. Ideally the phases of W should be the same as that of the original image. This is not always the case due to the rounding effects and the compression. However our results have shown that for the usual JPEG quality settings the phase is only slightly modified. This is extremely important. The watermark detector takes the DFT of the spatial realization of W and determines the magnitude selected bit plane values. The detector uses the shared encryption key k on the selected bit plane to decrypt it and arrives at $\mathbf{S}$'. It is well known that the correlation of two spatial images is given the IDFT of the product of the DFT of one image with the conjugate DFT of the other image. We wish to see how similar the watermarked image is to the original image, thus one would like to do an auto-correlation test.

Unfortunately, we do not have the original image; we only have the watermarked image. However, we have hidden and extracted (up to noise from rounding and compression) the BPOF of the original image in the watermarked image. Therefore, based upon this and the already mentioned references on Fourier optics we propose the following as our test for "autocorrelation."

(4) correlation(k,l) = IDFT$\{ (\mathbf{S}'(k,l)) * (H_{POF}(k,l)) \}$
Where $H_{POF}(u,v)$ is the POF of the watermarked image W. Other type correlation test might also work, but to deal with the compression of JPEG we need a test that has some elasticity. We feel that ours does and we show our results below.

3.1. Hiding additional information

We may be able to hide more information than simply a BPOF based signature. By taking the *XOR* of the BPOF with some additional information we an embed that as the watermark. Of course this additional information has to be pre-encoded in some sort of error correcting manner to survive rounding and compression errors. The extractor recovers the hidden information, uses *XOR*, and decodes the additional information. The use of additional information lets us do more than authentication (include tge date or time stamp the image, include instructions, include a new pseudo-random number for the next image transmission, etc). We will not address additional information more in this paper.

4. PERFORMANCE METRICS

To analyze the performance of the proposed embedding, we define a number of metrics [9] in the correlation detector as given by Eq. (4), which gives us a 2-dimensional "plane" of values.

Let P_{max} and P_{second} denote the highest and the second highest peaks of the correlation plane, respectively. Note that the second highest peak is calculated excluding a small 7x7 pixel area centered around the highest peak. Let μ be the average value of the whole correlation plane, and μ_{sl} is the average value of the sidelobe. Here the sidelobe is defined as the part of the correlation plane where the correlation values are less than 0.5* P_{max}. The total area of the sidelobe in terms of number of pixels is defined with another parameter called full-area-at-half-maximum (*FAHM*).

Peak-to-average-correlation-energy (PACE) is the ratio of the highest correlation peak energy to that of the average correlation energy. This is a measure of sharpness of the peak. We calculate this ratio in db by expressing it as

$$PACE = 20\log_{10}(\frac{P_{\max}}{\mu}) \ .$$

Next we calculate the peak-to-secondary-peak ratio (PSR), which is a measure of degree of false positive in a detection algorithm**.**

$$PSR = 20\log_{10}(\frac{P_{\max}}{P_{\text{second}}}) \ .$$

Higher PSR value indicates it is less likely to have a false positive.

The third metric is the output signal-to-noise ratio (SNR), which makes uses of P_{max} and sidelobe statistics. Let σ_{sl} represents the standard deviation of the sidelobe correlation values. In a good autocorrelation scenario, the sidelobe variance is expected to be small. The metric is defined as

$$SNR = 20\log_{10}(\frac{P_{\max} - \mu_{sl}}{\sigma_{sl}}) \ .$$

Finally, we explore another metric called the spatial resolution factor that measures how well separated the highest peak is spatially from the other correlation peaks. It is obtained as follows

$$SRF = 10\log_{10}(\frac{1}{FAHM}) \ .$$

5. RESULTS AND DISCUSSION

The simulation results that follow are based on 9 different 512x512 images selected from USC SIPI Database [8].

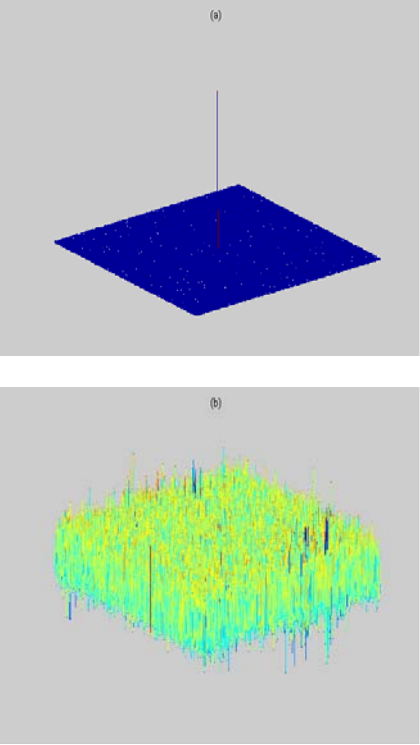

Figure 1

We adopt the standard practice of plotting using negative and positive frequencies (same for the correlation).

Figure 1(a) above shows the result of a typical correlation when an (TIFF) image is watermarked with the proposed BPOF signature. In contrast Fig. 1(b) shows the correlation output when the detector works on an un-watermarked original image. This shows a highly discriminatory signature embedded.

TABLE I (TIFF files) Correlation Statistics Marked Images

Image	**PACE**	**PSR**	**SNR**	**SRF**
Baboon	49.8	38.5	50.3	0
Bridge	49.7	38.5	50.1	0
Earth	50	39.1	50.6	0
Fishing_Boat	50	38	50.5	0
Lenna	50	37.8	50.6	0
Oakland	49.9	38.5	50.4	0
Peppers	50	38.8	50.6	0
Toy_vehicle	49.1	29.5	49.1	0
Water	49.9	38.5	50.5	0

TABLE II (TIFF files) Correlation Statistics Unmarked Images

Image	**PACE**	**PSR**	**SNR**	**SRF**
Baboon	14.3	1.4	12.6	-31.1
Bridge	13.7	0.4	12	-33.2
Earth	18.4	3.9	16.7	-11.8
Fishing_Boat	15.7	0	13.8	-26.3
Lenna	18.2	0.5	16	-20
Oakland	13.7	0.3	12	-33.2
Peppers	17.5	0.1	14.9	-24.8
Toy_vehicle	23.5	1.3	19.2	-15.7
Water	24.8	2.5	22.7	-7

TABLE III JPEG Compression Performance

Quality Factor	**PACE**	**PSR**	**SNR**	**SRF**
40	21.9	2.1	19.5	-11.8
50	25.9	6.4	23.8	0
60	28.9	10.4	26.9	0
70	33	11.4	31.2	0
80	37.8	13.6	35.9	0
90	44.8	20.1	43.1	0

Looking at Tables I&II shows us how the performance metrics truly can detect whether a TIFF image is watermarked or not. For space reasons we will not discuss further the values that can be used to make watermarking decisions, but we see clear separation in the results. Table III shows how our watermark can survive JPEG compression, to a point. However, using a standard quality level around 75% still validates our approach.

Much still has to be done but is beyond the scope of this introductory paper. Different bit planes and multiple bit planes may yield more robust watermarking. Also, the important issue of additional information needs to be included in the watermarking algorithm. An advantage of our method over more standard uses of an image hash as an authenticator is that our method works on generic JPEG files, without any additional payload being added on.

We also wish to investigate the use of partial bit planes. Since JPEG is so disruptive to high frequencies, a variant of our watermarking process that only uses mid-level frequencies may yield better results.

We also wish to study our method against the known watermarking attacks and see how our method can be improved.

7. ACKNOWLEDGEMENTS

Research funded by the Office of Naval Research.

8. REFERENCES

[1] F. Ahmed and M. A. Karim, "A Filter-feature-based Rotation Invariant Joint Fourier Transform Correlator," *Applied Optics*, Vol. 34, No. 32, pp. 7556-7560 (1995).

[2] R. Anderson, *Security Engineering*, Wiley, NY, 2001.

[3] I. Cox, J. Bloom, and M. Miller, "Digital Watermarking: Principles & Practice," 2001, Morgan Kauffman Publishers

[4] D.L. Flannery and J.L. Horner, "Fourier Optical Signal Processors", *Proc of the IEEE*, vol. 77, no. 10, pp. 1511-1527, 1989.

[5] J.L. Horner and P. D. Gianino, "Phase-only matched Filtering," Appl. Opt., vol. 23, pp. 812-816, 1984.

[6] J.L. Horner and J.R. Leger, "Pattern Recognition with Binary Phase-only Filters," *Applied Optics*, Vol. 24, No. 5, pp. 609-611, 1985.

[7] A. Oppenheim and J. Lim, "The importance of phase in signals," Proc. IEEE, vol. 69, pp. 529-541, May 1981.

[8] USC-SIPI Image Database, http://sipi.usc.edu/services/database/Database.html.

[9] B.V.K. Vijay Kumar and L. Hassebrook, "Performance Measures for Correlation Filters," Applied Optics, Vol. 29, no. 20, pp. 2997-3006.