

Phase-Signature Based Watermarking for Multimedia Authentication: Analysis and Design

Farid Ahmed^{*a}, Ira S. Moskowitz^b

^aThe Catholic University of America, Dept. of EECS, Washington, DC, USA 20064

^bNaval Research Laboratory, CHACS-Code 5540, Washington, DC USA 20375

ABSTRACT

In this work, we address phase-signature based digital image watermarking. The signature is extracted from the Fourier phase information of the digital media. It is then embedded in the Fourier magnitude spectrum. The detection and/or authentication is based on the well established area of phase-only filter based correlation techniques in the optics community. We propose to analyze the distortion coming out of the embedding process, model it, and eventually parameterize it so that optimal embedding can be done by trading off with other aspects of watermarking. It will be shown why permutation like functions facilitate the signature embedding process by minimizing the embedding degradation.

Keywords: Authentication, Watermarking, Correlation, Phasemark™, information hiding, phase-only filter.

1. INTRODUCTION

Multimedia authentication is concerned with the verification of the integrity of the contents and the ownership of digital multimedia data. A number of algorithms and protocols based on concepts such as cryptographic hash and digital signature are already in force to facilitate the authentication application in a networked environment while the data is in-transit. Recently, digital watermarking based techniques are also attracting increased interest due to their flexibility of ubiquitous security offered both during the in-transit and in-storage modes.

Digital watermarking, or more simply put, *watermarking*, is a technology for inserting information (the *watermark*) into digital media [1]. Watermarking is part of the larger topic of *information hiding*. The watermark should not disturb its original media (host or cover) “too much” that is, the integrity of the host must be preserved at some high level. Depending on the motivation, the watermark can be robust, semi-fragile or fragile (see below). Examples of some typical applications of watermarking include: copyright protection; authentication; fingerprinting; copy protection; and digital asset management in general [1]-[2]. In this paper, we primarily address the authentication of digital images (more simply put as “image”).

Authentication watermarks assist in deciding the authenticity of an image. The degree of authenticity depends on the fragility of the hidden information that is the watermark. If the watermark is trivially removable and is able to be replaced with a spoofed watermark, then we have no faith in the authenticity of the image. If the watermark can be removed or destroyed, but a substitute watermark cannot be inserted into the original “covering” or “cover” image, then we can put faith in the authenticity of the image if we do detect a watermark. A watermark is *fragile* if it “breaks” with any sort of image manipulation. Therefore, a fragile watermark is very sensitive and can be used both for authentication, and to detect tampering. In practice, we wish to tolerate some non-malicious acceptable modifications done to the image, such as JPEG compression. That is why authentication watermarks

* ahmed@cua.edu

are often *semi-fragile* in the sense that they are robust to some acceptable change in the image content or format, while at the same time they are fragile to undesirable tampering. Thus, by using a semi-fragile watermark, the authentication process is expected to detect tampering. We do not want to use a totally *robust* watermark because it would not detect content change.

Comparing an extracted media signature to a given “reference” signature is a standard means of watermark detection [1]. If the detector needs the original (un-watermarked) image for the “reference” signature, then it is called *non-blind* detection. In many practical applications, the original is usually not available, thus the obvious necessity for *blind* detection.

The *Phasemark* method that we are analyzing is a *blind, semi-fragile, signature-based* watermarking technique. It was first reported in [3], extended to wavelets in [4], and has been applied to biometrics in [5]. The purpose of this paper is to discuss in detail why Phasemark works and identify the design parameters.

2. PHASEMARK

Let us look at the Phasemark method reported in [3]. Figure 1 shows the flowchart of the watermark embedding process. The original image is transformed from the spatial domain to the frequency domain via the discrete Fourier transform (DFT). Consider an $M \times N$ (bit map representation of a grayscale image) host image $h(m,n)$, where (m,n) are the spatial indices (pixel locations). The DFT of $h(m,n)$ is written as $H(u,v)$ where (u,v) represent the frequency coordinates.

$$H(u,v) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} h(m,n) \exp(-j2\pi(\frac{um}{M} + \frac{vn}{N})) \quad (1)$$

Here j is the principal valued square root of -1. Note that sometimes in the literature the frequency components are normalized in the forward DFT. We do not use any normalization factor and thus achieve a greater flexibility of modifying the frequency components in the said watermarking process (of course the IDFT is normalized, as will be shown later). As seen from Eq. (1), the frequency components are complex numbers with real and imaginary parts. Switching to polar coordinates, the Fourier transform of the image can equivalently be expressed as

$$H(u,v) = X(u,v) \exp(j\phi(u,v)) \quad (2)$$

Where, $X(u,v)$ is the *magnitude* $|H(u,v)|$ of the frequency component, and $\phi(u,v)$ is the *phase* part of the (u,v) frequency given by

$$\phi(u,v) = \arg\left(\frac{\text{Re}(H(u,v))}{\text{Im}(H(u,v))}\right) \quad (3)$$

The magnitudes span a range of values from 0 to MAX, and the phases are in the interval $(-\pi, \pi]$.

In order to extract the signature, the *phase-only filter (POF)* is first obtained from Eq. [2] by first normalizing (setting the magnitude in all frequencies identically to 1), and then by taking the complex conjugate. Thus,

The POF:

$$H_{POF}(u,v) = \frac{\overline{H(u,v)}}{|H(u,v)|} = \exp(-j\phi(u,v)) \quad (4)$$

This “continuous” POF has values on a unit circle. We next “binarize” the POF by using a threshold. One widely used formulation [6] is to employ the imaginary axis as the threshold as given below.

The BPOF:

$$B_n(u, v) = +1, \text{ if } \cos(\phi(u, v)) \geq 0$$

$$= 0, \text{ otherwise}$$
(5)

We then encrypt this pattern to obtain the final signature $S(u, v)$. The encryption technique will be detailed in next section.

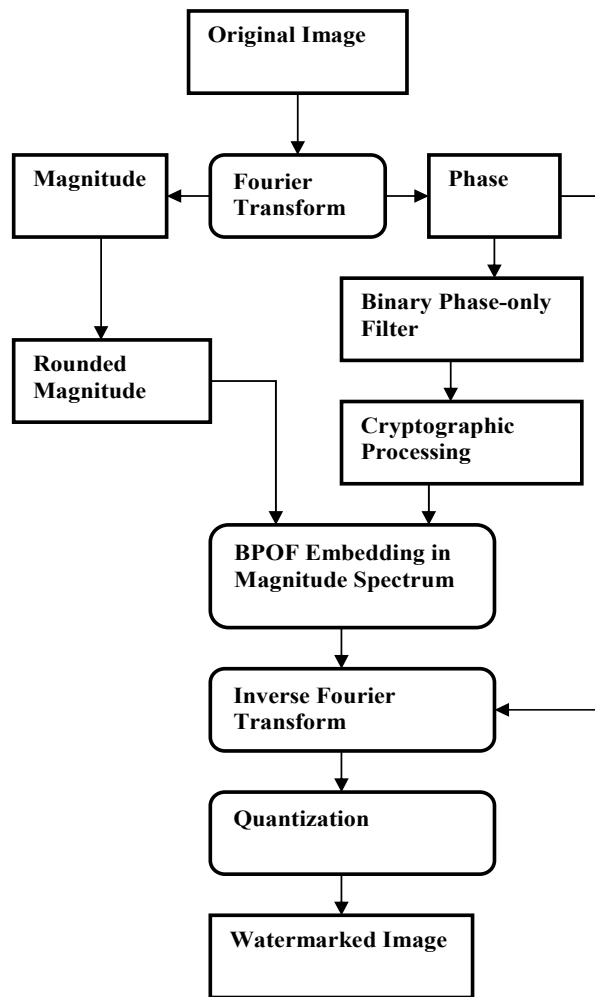


Fig. 1 Phasemark Embedding flowchart.

In the embedding process, the phase is kept unchanged and the magnitude $X(u,v)$ is modulated. The real-valued $X(u,v)$ is first transformed to integer-valued as follows.

$$R(u,v) = \text{round}[X(u,v)] \quad (6)$$

Where the $\text{round}(\cdot)$ function rounds the operand to the nearest integer value. That makes it representable by a fixed number of q bit planes (depending on what MAX is). Hence we have $R=R_{q-1}, R_{q-2}, \dots, R_1, R_0$ where R_i is the i -th bit plane of the rounded magnitude. This is an important pre-processing step to make it suitable for subsequent bit plane embedding.

After the bit-plane embedding (will be discussed in section 3), the Fourier spectrum becomes

$$\tilde{H}(u,v) = \tilde{R}(u,v) \exp(j\phi(u,v)) \quad (7)$$

To the above we apply the IDFT. The watermarked pre-image (in the spatial domain) is then given by

$$\tilde{h}(m,n) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \tilde{H}(u,v) \exp(j2\pi(\frac{um}{M} + \frac{vn}{N})) \quad (8)$$

This will result in a real-valued image if the Fourier spectrum matrix is Hermitian as noted earlier. We reiterate that we force this property to be maintained in the cryptographic processing. We now convert the watermarked pre-image to an actual grayscale bitmap in the spatial domain.

$$\tilde{h}_w(m,n) = \text{uint8}(\tilde{h}(m,n)) \quad (9)$$

Here the operation uint8 converts the operand to an integer in the range [0,255] (of course the values must be clamped at 0, or clipped at 255). Therefore, the values can be represented as unsigned integers in 8 bits, hence we have an “actual” spatial image.

The detection process includes the extraction of the signature and a subsequent correlation operation. We start with a test image $t(m,n)$. The discrete Fourier transform of this yields

$$T(u,v) = |T(u,v)| \exp(j\phi_T(u,v)) \quad (10)$$

After a rounding operation on Fourier magnitudes, we extract the desired bit plane and undo the cryptographic processing done during the embedding process. As an example, if the cryptographic processing is permutation, then the detector generates the inverse permutation matrix using the shared key. It then applies the permutation matrix on “half” of the extracted plane (because the permutation was done on half the BPOF plane). We then force the symmetry condition to come up with the extracted hidden signature S' . From the phase information of the test image, the POF is computed as

$$H_{POF}^T(u,v) = \exp(-j\phi_T(u,v)) \quad (11)$$

The next step is the correlation of the extracted and computed signals. It is well known that the correlation of two spatial images is given by 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 supposed 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 (if it is watermarked). Therefore, based upon this and the already mentioned references on Fourier optics we propose the following as our test for detection.

$$Corr(k,l) = IDFT(H_{POF}^T(u,v) \bullet S'(u,v)) \quad (12)$$

Next we will analyze the algorithm in terms of the suitability of BPOF as a signature, the bit-plane embedding, embedding error, ways to minimize embedding error, impact on bit-plane embedding on the quality of the watermarked image.

3. ANALYSIS OF PHASEMARK

3.1 BPOF as Signature

After the seminal work on the importance of phase [7], the BPOF was used extensively in the area of optical pattern recognition [8,9] due to its high discriminatory features. Since we use correlation as a measure of authentication, results from these works are equally applicable in our present work. In particular, the following characteristics make the BPOF a good choice as a signature in authentication applications, like ours.

- Phase have more discriminatory features when compared to magnitude.
- The POF and consequently the BPOF is better from the correlation point of view because the high spatial frequencies decorrelate very quickly. On the other hand, emphasizing high frequencies does make it more vulnerable to the noise and scale and rotation changes.
- It will be a good signature for most of the images except where the phase distribution is identical. This includes all images with positive real DFT. One example is images whose intensity is Gaussian distributed. But fortunately no real, natural image has this property.
- The phase quantization in BPOF has built in tolerance, at least up to half the quantization level, against some desirable tampering.
- Binary patterns are well suited for embedding.

3.2 Bit-plane Embedding

The core of our watermarking process is the bit plane embedding. From the rounded magnitude spectrum of the original image $R(u,v)$, we first select i^{th} (the choice of i is discussed later) bit plane and replace it after modulating it with the cryptographically processed phase-based signature ($S(u,v)$), as follows

$$\tilde{R}_i(u,v) = C(u,v) \bullet S(u,v) \quad (13)$$

Here \bullet denotes a logical operation of the signature bit plane and any other bit plane $C(u,v)$ (not necessarily from the host image). If the operator is XOR and $C(u,v)$ is an all-zero bit plane, then the embedding is simply equivalent to replacing the selected bit plane with the signature plane. In another scenario, if the operation is XOR and $C(u,v) = R_i(u,v)$ and furthermore the signature does not degrade much, then Eq. (13) can be used to retrieve the original bit plane and thus the original image. This is an example of a reversible watermarking. In yet another realization, $C(u,v)$ may represent any bit plane other than the embedding plane, which gives an additional degree of security. Let us now look at the bit plane embedding little closely. In the selected bit plane the following transition occurs

Prior Bit	New Bit	Magnitude Change
0	0	0
0	1	$2^{(i-1)}$
1	0	$-2^{(i-1)}$
1	1	0

We are interested in the transitions $0 \rightarrow 1$ and $1 \rightarrow 0$. For the $0 \rightarrow 1$, transition, it is true that all frequencies having magnitude less than 2048 (for $i = 12$) have an old bit set to 0. Therefore it enhances the high frequencies, whose magnitude originally was smaller. The $0 \rightarrow 1$ transition also happens for a number of high magnitude frequencies. But the relative change is smaller than that in high frequencies. For the $1 \rightarrow 0$ transition, the coefficients which had a 1 before are part of low frequencies, which is now decremented in value with this transition. Therefore, the overall change is to relatively increase the magnitude of the high frequencies.

3.3 Analysis of Embedding Error

Now let us look at how much error is introduced because of the watermarking process itself. Keep in mind that the values of $\tilde{h}(m,n)$ have not been clipped, clamped, or rounded into integers between 0 and 255. This is why we call $\tilde{h}(m,n)$ the pre-image.

$$\begin{aligned}
e(m,n) &= \tilde{h}(m,n) - h(m,n) \\
&= \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} (\tilde{H}(u,v) - H(u,v)) \exp(j2\pi(\frac{um}{M} + \frac{vn}{N})) \\
&= \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} (\tilde{R}(u,v) - R(u,v)) \exp(j\phi(u,v)) \exp(j2\pi(\frac{um}{M} + \frac{vn}{N})) \\
&= \frac{2^{i-1}}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} (\tilde{R}_i(u,v) - R_i(u,v)) \exp(j\phi(u,v)) \exp(j2\pi(\frac{um}{M} + \frac{vn}{N}))
\end{aligned} \tag{14}$$

Let us concentrate on the term $e(0,0)$, this is the difference between the original image and the pre-image only at the (0,0) pixel.

$$e(0,0) = \frac{2^{i-1}}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} (\tilde{R}_i(u,v) - R_i(u,v)) \exp(j\phi(u,v)) \tag{15}$$

Since $e(0,0)$ is a real number (also the symmetries of the IDFT cause this) we see that the above can be re-written as

$$e(0,0) = \frac{2^{i-1}}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} (\tilde{R}_i(u,v) - R_i(u,v)) \cos(j\phi(u,v)) \tag{16}$$

If we do not permute the BPOF the term $e(0,0)$ is quite large. The reason for this is as follows, in the above equation

$$\cos \phi(u,v) \geq 0 \text{ iff } \tilde{R}_i(u,v) = 1, \text{ and } \cos \phi(u,v) < 0 \text{ iff } \tilde{R}_i(u,v) = 0.$$

Therefore we only sum terms that are 0 or 1. Therefore, we see that $e(0,0)$ is $O(2^{i-1})$. This has two bad effects, one is that the (0,0) pixel of the pre-image is very different from the (0,0) pixel of the original image. The other is that if one has the pre-image they can easily destroy the watermark by adjusting the (0,0) pixel. Of course, keep in mind that the final watermarked image is not the pre-image and these differences are lessened.

Now, let us see what the total difference between the pre-image and the original image is. This is given by

$$\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e(m,n) = \frac{2^{i-1}}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} (\tilde{R}_i(u,v) - R_i(u,v)) \exp(j\phi(u,v)) \exp(j2\pi(\frac{um}{M} + \frac{vn}{N}))$$

which we also write as

$$\begin{aligned}
& \frac{2^{i-1}}{MN} \sum_{u,v} (\tilde{R}_i(u,v) - R_i(u,v)) \exp(j\phi(u,v)) \sum_{m,n} \exp(j2\pi(\frac{um}{M} + \frac{vn}{N})) \\
&= \frac{2^{i-1}}{MN} (\tilde{R}_i(0,0) - R_i(0,0)) \exp(j\phi(0,0)) \sum_{m,n} 1 \\
&+ \frac{2^{i-1}}{MN} \sum_{u>0,v>0} (\tilde{R}_i(u,v) - R_i(u,v)) \exp(j\phi(u,v)) \sum_{m,n} \exp(j2\pi(\frac{um}{M} + \frac{vn}{N})) \\
&= 2^{i-1} (\tilde{R}_i(0,0) - R_i(0,0)) \exp(j\phi(0,0)) \\
&+ \frac{2^{i-1}}{MN} \sum_{u>0,v>0} (\tilde{R}_i(u,v) - R_i(u,v)) \exp(j\phi(u,v)) \sum_{m,n} \exp(j2\pi(\frac{um}{M} + \frac{vn}{N}))
\end{aligned}$$

Since the DC frequency is a real number we have that $\exp(j\phi(0,0)) = 1$, so the above reduces to

$$2^{i-1} (\tilde{R}_i(0,0) - R_i(0,0)) + \frac{2^{i-1}}{MN} \sum_{u>0,v>0} (\tilde{R}_i(u,v) - R_i(u,v)) \exp(j\phi(u,v)) \sum_{m,n} \exp(j2\pi(\frac{um}{M} + \frac{vn}{N}))$$

which is also

$$\begin{aligned}
& 2^{i-1} (\tilde{R}_i(0,0) - R_i(0,0)) \\
&+ \frac{2^{i-1}}{MN} \sum_{u>0,v>0} (\tilde{R}_i(u,v) - R_i(u,v)) \exp(j\phi(u,v)) \sum_m \exp(j2\pi(\frac{um}{M})) \sum_n \exp(j2\pi(\frac{vn}{N}))
\end{aligned} \tag{17}$$

Consider $\sum_m \exp(j2\pi(\frac{um}{M}))$, if u and M are relatively prime, then this sum is just the sum of the M distinct roots of unity, which sum to 0. If they are not relatively prime, say $M = bu$ (b a positive integer greater than 1), then the sum is just the sum of the b distinct roots of unity, u times, which is also 0. Of course $\sum_n \exp(j2\pi(\frac{vn}{N}))$ behaves the same. Therefore we have that

$$\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e(m,n) = 2^{i-1} (\tilde{R}_i(0,0) - R_i(0,0)) \tag{18}$$

If the BPOF is not permuted then $\tilde{R}_i(0,0) = 1$, since $\exp(j\phi(0,0)) = 1 = \cos \phi(0,0)$. If the BPOF is permuted then $\tilde{R}_i(0,0)$ can be 0 or 1. This tells us that the difference between the pre-image and the original image is 2^{i-1} times (1 or 0) if the BPOF is not permuted, and is 2^{i-1} times (1 or 0 or -1) if the BPOF is permuted.

3.4 Reducing Embedding Error: Encrypting the Signature

As discussed, in absence of a cryptographic tweak, there is a prohibitively large rounding error on the (0,0) element of watermarked image. The reason for this high error at the DC value is due to the

strong correlation of the BPOF signature with its own Fourier magnitude spectrum. The general treatment for this error is to randomize the signature pattern before embedding. We use an encryption method that serves a dual purpose – reduces the embedding error as well as it protects the authentication against forgery attack.

A problem with off-the-shelf public or symmetric key encryption is that even a small change in the encrypted signature may have an *avalanche* effect [10] in the decrypted version, thus making the authentication too fragile to use. Specifically, our concern is that any noise that comes about from our watermarking or from compression affects the pixels in a sparse manner. We do not want the noise pixels to affect the non-noisy pixels. Therefore, our encryption must treat each pixel coordinate independent of the others. Hence, this is why we do not use stream ciphers. Since our objective here is a semi-fragile watermark, we use a very rudimentary form of key-based permutation, as a proof of concept, which does not, by design, have the avalanche effect. (We envision using public key cryptography in future work).

We permute the values of BPOF via a cryptographic key $\mathbf{S}(u,v) = E_k[B_n(u,v)]$ (or more simply expressed as \mathbf{S}). Here E_k may represent a variety of key(k)-based cryptographic techniques and protocols that may be used in this regard. This is done to prevent someone from spoofing a watermark, but (as discussed *ad nauseam*) it also has the desirable consequence of modifying the errors in the $e(0,0)$ term. This is clear from eq. (16), because: $\cos \phi(u,v) \geq 0$ iff $\tilde{R}_i(u,v) = 1$, and $\cos \phi(u,v) < 0$ iff $\tilde{R}_i(u,v) = 0$, no longer holds! Experimentation has shown that after encryption the term $e(0,0)$, in the pre-image, is no longer an outlier.

Note that $B_n(u,v)$ inherits a symmetry ($B_n(M-u,N-v) = B_n(u,v)$) from the fact that the DFT of an image is “Hermitian” in the sense that $H(u,v) = H^*(M-u,N-v)$, where $*$ is the complex conjugate. (It is a common technique to view the frequencies as both positive and negative frequencies. The (0,0) frequency “becomes” the origin, but the frequencies ($M-u,N-v$) are taken as frequencies ($-u,-v$). This is usually obvious in plots of Fourier magnitudes where one obtains true symmetries between the upper right (left) coordinates and the lower left (right) coordinates.) Therefore, all encryption discussed here must preserve this symmetry.

In our case, we simply use a key as a seed for a pseudo-random number generator that determines a permutation of the frequencies. To preserve the symmetry, in fact we only permute “half” the frequencies (we will not go into implementation details of this) and then extend it to the other half. Since there are so many frequencies we feel that this is very difficult to decipher (at least in one use of a given permutation). This permutation operation results in the actual signature \mathbf{S} , which is subsequently embedded.

3.5 Perceptual Quality of the Watermarked Image

We choose one bit-plane from the rounded Fourier magnitude depending on the constraint of image quality degradation. Generally, one of the mid-level bit planes works best. Experiments show that an optimal value of selected bit plane i is given by

$$i = \lceil q/2 \rceil$$

where q is the maximum bits required to represent the Fourier magnitude. For an $M \times N$ image having 256 gray levels (representable in 8 bits), q is given by $q = \log_2(M \times N) + 8$. Therefore for a 256×256 image, $q=24$ and the optimal bit plane is 12 (which is what we used in our previous section).

Now, for a selected bit-plane there are two sources of quality degradation of the original image in our watermarking process. The first one is coming out from the very embedding process as encapsulated

by eq. (14). The second one is due to the fact that the watermarked image which is the real part of the IDFT of the modified spectrum, contains non-integer values. This need to be converted to 8-bit integer values as shown in eq. (9). During this rounding process some information of the hidden image is lost. Considering these two, we use the metric *peak signal-to-noise ratio* (PSNR) to benchmark the quality of the watermarked image in dB scale as follows.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{\left[\frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} ((h(m,n) - \tilde{h}_w(m,n))^2) \right]} \right) \quad (19)$$

Now, combining eq. (14) and eq. (9) with the above, we can say that error almost doubles for increasing the bit plane by 1, which has the same effect of PSNR increasing by approximately 6 dB. Following table shows PSNR values of different “standard” images [3,11] at different embedding strengths.

TABLE III
QUALITY OF THE MARKED IMAGES

Image	PSNR (dB)		
	i=12	i=13	i=14
Baboon	38.93	33.03	27.04
Bridge	38.92	33.07	27.1
Earth	38.91	33	26.99
Fishing Boat	38.93	33.04	27.1
Lenna	38.9	32.97	26.98
Oakland	38.91	33.02	27.05
Peppers	38.90	32.99	27.05
Toy vehicle	38.95	33.1	27.19
Water	38.96	33.07	27.1

Another interesting observation is that increasing the number of bit-planes for embedding does not significantly change the quality of the marked image. This allows us to embed additional information, which will be pursued later.

4. CONCLUSION

In analyzing the Phasemark method, we have identified the source of errors in the phase-based bit-plane embedding technique and found an effective way of minimizing this error. We have also shown how the perceptual quality of the watermarked image depends on the choice of bit plane and also proposed an optimal choice of the bit plane.

Further works can be done on optimal Binary Phase-only filter in terms of Quantization Error and correlation

- a) Increasing robustness to attacks by developing a region of support for the BPOF.
- b) Increasing robustness by multiple bit-plane embedding.

ACKNOWLEDGEMENTS: We thank LiWu Chang for his helpful discussions.

REFERENCES

- [1] Ingemar Cox, Jeffrey Bloom, Matthew Miller, *Digital Watermarking: Principles & Practice*, 2001, Morgan Kauffman Publishers, ISBN 1-55860-714-5, ch. 1-2.
- [2] Digimarc Corporation. www.digimarc.com.
- [3] Farid Ahmed and Ira S. Moskowitz, "Correlation-Based Watermarking Method for Image Authentication Applications," *Optical Engineering*, Vol. 43, No. 8, pp. 1834-1838, August 2004.
- [4] Farid Ahmed and Ira S. Moskowitz, "Phase Signature-based Image Authentication Watermark Robust to Compression and Coding," Proc. SPIE 49th annual meeting, Conference 5561, Mathematics of Data/Image Coding, Compression, and Encryption VII, with Applications, Denver, August 4-5, 2004.
- [5] Farid Ahmed and Ira S. Moskowitz, "Composite Signature Based Watermarking for Fingerprint Authentication" Proc. ACM Multimedia and Security Workshop, August 1-2, 2005, NY, NY.
- [6] David L. Flannery and Joseph L. Horner, "Fourier Optical Signal Processors", *Proc of the IEEE*, vol. 77, no. 10, 1989.
- [7] A. Oppenheim and J. Lim, "The importance of phase in signals," *Proc. IEEE*, vol. 69, pp. 529-541, May 1981.
- [8] D. Psaltis, E. Paek, and S. Venkatesh, "Optical Image correlation with binary spatial light modulator," *Opt. Eng.*, pp. 698-704, 1984.
- [9] J. L. Horner and J. R. Leger, "Pattern Recognition with Binary phase-only filters," *Appl. Opt.*, vol. 24, pp. 609-611, 1985.
- [10] William Stallings, *Cryptography and Network Security-Principles and Practices*, 3rd ed. Prentice Hall, 2003, ch. 3.
- [11] USC-SIPI Image Database, <http://sipi.usc.edu/services/database/Database.html>.