Volume 4, Issue 5, May 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Robust Watermarking Of Greyscale Images In Frequency Domain Using M Sequences Correlation Properties

Vinod Todwal[1], Geeta Gupta[2],Sachin Sharma[3]
[1]Department of Computer Science, RIET, Jaipur
[2]Department of Information Technology, RCEW, Jaipur
[3] Department of Computer Science,RIET,Jaipur

Abstract— ***Proliferation of internet, together with relatively inexpensive digital recording and storage peripherals has created an era where duplication, unauthorized use and maldistribution of digital content has become easier. Thus, protection of Intellectual Property Rights (IPS) is of great importance to prevent unauthorized use, misappropriation and misrepresentation. Digital watermarks are pieces of information added to digital data (audio, video, or still images) that can be detected or extracted later to make an assertion about the data, thereby providing protection for IPS. This information can be textual data about the author, its copyright, etc. or it can be an image itself. This watermark in a document assert the ownership of the document. In case of image watermarking, it is often the requirement that watermark should be invisible to human. Also, unauthorized removal and detection of the watermark must be impossible even if the watermarking scheme is partially known. A digital watermark in an image can be embedded in spacial (pixels values) or in transform (Discrete Fourier, Discrete Cosine, etc.) domain. In this work, a technique for watermarking of grayscale images in frequency domain is presented. A phase-based signature of the image, modulated by PN (Pseudo Noise) sequence, is hided into its Fourier magnitude spectrum in the embedding stage. The detector computes the Fourier transform of the watermarked image and extracts the embedded signature. The proposed method provides blind detection at the receiver which is a major advantage, due to the fact that neither original data knowledge nor time consuming search in owners' database is needed for watermark detection and extraction. Experimental results indicate the ability of the proposed method to deal with the a large class of attacks. Results are compared with existing techniques and conclusions are provided.***
Keywords— ***Watermarking, Greyscale Images, Fourier Transform, PN sequences, M sequences, Spread Spectrum***

I. BPOF as watermark

Binary phase-only filter (BPOF), is an inherent characteristic of an image, and this can serve as the basis for watermark [23] One can start with the spatial realization of an image, which is the luminance values of the pixels (bitmap). Then the 2-dimensional Discrete Fourier transform (DFT) to the bitmap or a single 1 D DFT to the complete pixel array taken row wise can be applied . The watermarking process hides into the image, the BPOF, thus forming the watermarked image. This hiding is done by changing, in a specific manner, the magnitude of the Fourier coefficients, based on 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 such that these cannot be lost due to noise or compression. Then the inverse transform is applied to the image, thereby getting back the watermarked image. This technique can be attacked so that the watermark can be removed and hence a false negative can be achieved, but in the proposed work, the matching of the extracted information to the BPOF is accomplished using correlation metrics thereby providing robust watermarking.

II. Modulated BPOF with m sequence as a watermark

The process of embedding the watermark in frequency domain is accomplished by first converting the image pixel values to Fast Fourier Transform. FFT is a reversible transformation which provides perfect reconstruction of the original values. As per the requirement of the FFT algorithm to process the data samples of size in the power of two, it is the requirement of the algorithm that the number of pixels in the host image must be a power of two. This includes images of dimension 64X64, 128*128, 128X64 and likewise. Nevertheless, the scheme presented in the current work can be applied to watermark image of any dimension. The Binary Phase only Filter (BPOF) of the original Image is obtained using the phase values from the Fast Fourier Transform of the original image. This BPOF is then modulated by PN sequence which is used as the digital watermark of the original image. The process of embedding watermark using FFT in grayscale images is described in the following algorithm.

Algorithm: Embedding

1. Let the given image has n pixels, where $n = 2^k$ for some $k \in I^+$. Tabulate the pixel values of grayscale image row-wise starting from the top left corner. These values belongs to the range from 0 to 255, giving a total of 256 colors giving shades of grey from 0 as black to 255 as white covering all shades of grey in between these limits.
2. Obtain the FFT of the data values. These values are complex numbers of the form a+ib, where i is $\sqrt{-1}$. Convert these values in the form $Ae^{i\theta}$, where A is the magnitude and $e^{i\theta} = \cos\theta + i \sin\theta$.

3. Calculate the Binary Phase Only Filter (BPOF) as per the following scheme. For any sample value n, if cosθ>1, BPOF (n) = 1, else 0.
4. Modulate the BPOF using PN sequence obtained through LFSR. This BPOF is a bipolar binary image of the original image, which has been shown to possess a number of discriminatory features of the original image.
5. Using the Logical Reversible Operation XOR, operate the modulated BPOF with m^{th} bit plane and replace k^{th} bit pane with the resultant.
6. Obtain the Inverse DFT of the sample values and get the pixel values of the watermarked image.

This scheme is illustrated with the following example. Consider the pixel value matrix of any image as shown in figure 1.

251	200	255	244	199	201	160	177
121	199	201	160	177	201	160	200
188	150	200	199	177	201	160	200
199	144	159	177	201	160	177	177
177	201	160	200	232	199	180	200
200	200	130	199	177	201	160	200
199	270	177	201	160	200	188	129
121	177	201	160	200	199	198	133

Figure 1 An arbitrary portion of grayscale image of 8X8 pixels

Given a sequence of N samples f(n), indexed by n = 0..N-1, the Discrete Fourier Transform is defined as F(k), where k=0..N-1:

$F(k) = [1/\sqrt{N}]*SUM_{k=0}^{N-1}f(n)e^{-i2\pi kn/N}$

F(k) are often called the 'Fourier Coefficients' or 'Harmonics'.

The sequence f(n) can be calculated from F(k) using the Inverse Discrete Fourier Transform (IDFT)

$f(n) = [1/\sqrt{N}]*SUM_{k=0}^{N-1}F(k)e^{i2\pi kn/N}$

The factor $[1/\sqrt{N}]$ is sometimes skipped for the sake of fast computations. The frequency domain representation, using DFT, of the pixel values, taken row wise, starting from top left corner, is shown in the figure 2.

11904	84.5989735532825+91.58609007891 64i	109.954253778935-196.770687987655i	265.882981467977-82.4839846089861i	388.704693977262-290.277682483071i	249.937523415086+116.407771595575i	-94.11407674321 81-54.7928852122747i	-103.549604053545-289.195840039834i
-169.5304832720 5-172.8319984622 14i	-62.7818773779053-157.598404017927i	-165.859251629507-91.6318751899344i	-38.8194317278381-93.9210838634986i	38.0352385189684-71.1398668004932i	154.604760862055+124.053609268126i	35.2242495404475+90.5396868112047i	-437.318558780238-218.105033079495i
113-147i	-77.70361474352-125.005815630618i	58.2655655477258-59.3066727265793i	205.138478050536+114.744534856005i	157.229696744737+35.79620806257 74i	317.343506999774+38.8792286502638i	16.0789612804672-159.193242731837i	-165.977577592822+99.1305566378199i
35.5304832720493+27.1	335.916090261481+28.2018	9.08614250304412	200.822598489014+63.7574	112.030370759032	125.215699613699-194.900539	359.364155722103	130.690051562962+15

680015377854i	766086937i	+6.58182210889323i	576718133i	-131.34160762i	708124i	-209.680972662368i	1.697209271085i
-214	130.690051562962-151.697209271085i	359.364155722104+209.680972662367i	125.2156996137+194.900539708124i	112.030370759032+131.34160762i	200.822598489014-63.7574576718135i	9.0861425030445-6.58182210889345i	335.916090261481-28.2018766086949i
35.5304832720494-27.1680015377857i	-165.977577592822-99.1305566378193i	16.0789612804679+159.193242731837i	317.343506999775-38.8792286502647i	157.229696744737-35.7962080625776i	205.138478050536-114.744534856006i	58.2655655477261+59.3066727265792i	-77.7036147435187+125.005815630618i
113+147i	-437.318558780237+218.105033079497i	35.2242495404475-90.5396868112044i	154.604760862054-124.053609268126i	38.0352385189683+71.1398668004935i	-38.8194317278379+93.921083863499i	-165.859251629506+91.631875189935i	-62.7818773779047+157.598404017927i
-169.530483272049+172.831998462215i	-103.549604053544+289.195840039835i	-94.114076743218+54.7928852122752i	249.937523415086-116.407771595575i	388.704693977262+290.277682483071i	265.882981467977+82.4839846089856i	109.954253778936+196.770687987655i	84.5989735532823-91.5860900789163i

Figure 2 FFT sample values of the pixel values of fig 3.1

It is clear from the table that the first entry is the sum of all the values of individual pixels, the second entry is the complex conjugate of the last entry and likewise.

III. Computing (BPOF) from FFT

The Binary Phase only Filter (BPOF) signature can be obtained by converting the complex sample values to the form $Ae^{i\theta}$ and computing the phase of the sample values. It is worth mentioning here that

$$e^{i\theta} = \cos\theta + i\sin\theta.$$

BPOF sample value is 0 if $\cos\theta < 1$, and 1 otherwise. BPOF possess several properties of the original image and can itself be used as a watermark or a digital signature. The BPOF signature for the above sample values can be obtained as shown in figure 3.

1	**1**	**1**	**1**	**1**	**1**	**0**	**0**
0	**0**	**0**	**0**	**1**	**1**	**1**	**0**
1	**0**	**1**	**1**	**1**	**1**	**1**	**0**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**
0	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**0**	**1**	**1**	**1**	**1**	**1**	**0**
1	**0**	**1**	**1**	**1**	**0**	**0**	**0**
0	**0**	**0**	**1**	**1**	**1**	**1**	**1**

Figure 3 BPOF signature corresponding to FFT sample values fig 3.2

It is clear from the above table that like the Discrete Fourier Transform of the original image pixel values, the BPOF also follows the same pattern that except, the starting value of the BPOF, the rest of the values follows after half of the sample values in reverse order. Thus, for a total of 64 sample values, BPOF of first 33 values can give the BPOF of the entire

pixel values. This result is straightforward as out of the total 2^n sample values of the DFT, two values are real numbers and the rest of the sample values can be partitioned into two parts with each value being complex conjugate of the other. This similar result hold for BPOF which can be obtained by diving the real part of the complex number with the absolute value of the complex number. Thus, the first 2^{n-1}+1 values of BPOF can give the entire sequence of the BPOF.
For a table consisting of 2^n values, the BPOF (and also the DFT) values form the pattern as shown in Table 1.

TABLE 1 CONJUGATE PROPERTY OF FFT AND CORRESPONDING EFFECT ON BPOF

Sample Data Values	**DFT**	**Analysis**
Value 1 (v1)	Sum(v1,v2..vn)	Only these first $2^{n/2}$ +1 values of DFT are sufficient to generate the entire sequence. Same is the case with BPOF.
Value 2 (v2)	Complex Number 1 (c1)	
. Value 3 (v3)	Complex Number 2 (c2)	
.		
.		
.		
Value 2^{n-1} + 1.(v2^{n-1} + 1)	Real Number 2^{n-1} + 1 (c $2^{n/2}$ + 1) (obtained by summing all the sample values, with negatives of alternate values starting from the second value.)	
.		
.		
.		
Value 2^n -1(v2^n - 1)	Complement of Complex Number 2 (c2)	
Value 2^n (v2^n)	Complement of Complex Number 1 (c1)	

Thus, for a total of 2^n values of BPOF, the first 2^{n-1}+1 values can generate the entire sequence as they are repeated again in the subsequent values in reverse order.
Consider only the first 2^{n-1} +1 values of the BPOF. Further, considering only the values except first and the last, these are total 2^{n-1}-1 values, which is equal to the length of code generated by an LFSR circuit having n-1 registers.

IV. Watermark Embedding

In the current scheme of 8X8 block size, watermarking procedure is operated on DFT samples from 2nd to 32nd sample, as it can be inverted to obtain the second half of DFT values.
The magnitude of the corresponding complex numbers can be tabulated as shown in figure 4.

101110 100000 00	000000 011111 01	00000 01110 0001	00000 10001 0110	000 001 111 001 01	00000 10001 0100	000 000 011 011 01	00000 10011 0011
000000 111100 10	000000 101010 10	00000 01011 1101	00000 00110 0110	000 000 010 100 01	00000 01100 0110	000 000 011 000 01	00000 11110 1001
000000 101110 01	000000 100100 11	00000 00101 0011	00000 01110 1011	000 000 101 000 01	00000 10100 0000	000 000 101 000 00	00000 01100 0001
000000 001011 01	000001 010100 01	00000 00000 1011	00000 01101 0011	000 000 101 011 01	00000 01110 1000	000 001 101 000 00	00000 01100 1000
000000 110101 10	000000 110010 00	00000 11010 0000	00000 01110 1000	000 000 101 011 01	00000 01101 0011	000 000 000 010 11	00000 10101 0001
000000 001011 01	000000 110000 01	00000 01010 0000	00000 10100 0000	000 000 101 000	00000 01110 1011	000 000 010 100	00000 01001 0011

				01		11	
000000 101110 01	000001 111010 01	00000 00110 0001	00000 01100 0110	000 000 010 100 01	00000 00110 0110	000 000 101 111 01	00000 01010 1010
000000 111100 10	000001 001100 11	00000 00110 1101	00000 10001 0100	000 001 111 001 01	00000 10001 0110	000 000 111 000 01	00000 00111 1101

Figure 4 Magnitude of the complex sample values of figure 3.3

Consider the m sequence of length 31, 1010111011000111110011010010000.
The 31 consecutive values of the BPOF, starting from the beginning and except the first value are as follows: 1111100000011101011110111111111
The Bitwise Exclusive OR operation on m sequence with the 31 bits (excluding first) of BPOF, yields the sequence as shown in figure 3.5.
The XOR operation is a reversible operation which can produce any of the inputs by again operating the output with any other input.
This Output is then substituted by the n^{th} bit of the input, depending upon the required robustness
Replacing the 4^{th} bit plane with the XOR output for all 64 values of DFT, one can have the following values as shown in figure 6.

10111 01000 0000 (Intact)	0000 0001 1101 01	0000 0011 1010 01	00000 10001 0110	0000 0111 1011 01	0000 0100 0101 00	0000 0001 1011 01	00000 10011 1011
00000 01111 0010	0000 0010 1010 10	0000 0010 1111 01	00000 00110 0110	0000 0001 0110 01	0000 0011 0011 10	0000 0001 1000 01	00000 11110 1001
00000 01011 0001	0000 0010 0110 11	0000 0001 0100 11	00000 01110 1011	0000 0010 1010 01	0000 0101 0000 00	0000 0010 1000 00	00000 01100 0001
00000 00010 0101	0000 0101 0110 01	0000 0000 0010 11	00000 01101 0011	0000 0010 1011 01	0000 0011 1010 00	0000 0110 1010 00	00000 01100 1000
00000 01101 0110 (Intact)	0000 0011 0010 00	0000 0110 1010 00	00000 01110 1000	0000 0010 1011 01	0000 0011 0100 11	0000 0000 0010 11	00000 10101 1001
00000 00010 0101	0000 0011 0000 01	0000 0010 1000 00	00000 10100 0000	0000 0010 1010 01	0000 0011 1010 11	0000 0001 0100 11	00000 01001 1011
00000 01011 0001	0000 0111 1010 01	0000 0001 1000 01	00000 01100 1110	0000 0001 0110 01	0000 0001 1001 10	0000 0010 1111 01	00000 01010 1010

00000011110010	00000100111011	00000001101101	00000100010100	00000111101101	00000100010110	00000011101001	00000001110101

Figure 6 Fourth Bit substituted with the XOR of M sequence and BPOF

.

The modified Fourier Transform after the substitution of the code (EX-OR of PN and BPOF) is as shown in figure 7.

11904	79.3881382147828+85.9448864726142i	113.657778858472-203.398400448378i	265.516711590291-82.3703579346237i	395.009301465237-294.98585009445i	250.194604691514+116.527506552131i	-94.1984749070084-54.8420215271051i	-106.187280725886-296.562407280562i
-169.461796280955-172.761973828826i	-62.9139162921922-157.929855115425i	-165.432127780595-91.3959030700218i	-38.9617758914151-94.2654762857916i	41.9630512677076-78.4863193703447i	160.671335142706+128.921379388604i	35.1697510963196+90.3996051309016i	-437.596510794002-218.243656817133i
107.8726645249-140.329926417348i	-81.8277355912678-131.640501700676i	58.1678811238511-59.2072428471512i	205.095581524419+114.720540615709i	164.783356301682+37.5159364264452i	317.625127341672+38.9137312678492i	16.0786404873629-159.190066650147i	-165.696627673254+98.9627585393171i
29.3921850372419+22.4744178731404i	343.790530128937+28.8629761747541i	8.90833780926303+6.4530239017111i	201.107992387012+63.8480649516131i	112.269912383164-131.62244023523i	125.400834414869-195.188705431576i	366.219240425917-213.680761749542i	130.540126039018+151.523184673889i
-214	130.540126039018-151.523184673889i	366.219240425918+213.68076174954i	125.40083441487+195.188705431576i	112.269912383164+131.62244023523i	201.107992387012-63.8480649516133i	8.90833780926305-6.45302390171107i	343.790530128937-28.8629761747553i
29.3921850372418-22.4744178731405i	-165.696627673254-98.9627585393167i	16.0786404873636+159.190066650147i	317.625127341672-38.9137312678285i	164.783356301681-37.515936426	205.095581524418-114.72054061571i	58.1678811238513+59.207242847151i	-81.8277355912668+131.640501700676i

				4454i			
107.8726645249+140.329926417348i	-437.596510794001+218.243656817135i	35.1697510963197-90.3996051309016i	160.671335142705-128.921379388604i	41.9630512677074+78.4863193703448i	-38.9617758914147+94.2654762857918i	-165.432127780595+91.3959030700227i	-62.9139162921917+157.929855115425i
-169.461796280954+172.761973828827i	-106.187280725885+296.562407280562i	-94.1984749070838+54.8420215271056i	250.194604691514-116.527506552131i	395.009301465237+294.98585009445i	265.516711590291+82.3703579346233i	113.657778858473+203.398400448377i	79.3881382147827-85.9448864726143i

Figure 7 Modified Fourier Coefficients with Fourth Bit modified.

The inverse Fourier Transform for the same is:

252	200	256	244	199	201	160	177
120	199	201	160	177	202	160	201
188	150	200	200	177	201	160	201
198	143	159	176	202	160	177	178
177	202	161	201	232	198	181	201
199	200	129	199	177	200	160	200
199	271	177	201	159	199	189	128
120	176	201	160	200	198	198	132

Figure 8 Modified (watermarked) pixel block of size 8X8.

The above figure gives the modified watermarked image block.

V. Watermark Detection

The bit sequence extracted from the fourth bit plane is operated bitwise XOR with BPOF to give the m sequence. This m sequence gives perfect cross correlation giving -1. This is the characteristic property of the m sequence and can be illustrated as follows:

For the calculation of the PN sequence correlation analysis, 0 is replaced by -1, thus giving a sequence of 1 and -1 and the sequence is known as chip rate.

The cross correlation of the extracted sequence with PN sequence generated through the same seed and same recurrence relation is given by:

Figure 3.10 Correlation Based Detection of Watermark

The mean square error of the above image block is 30. Also

PSNR = 20 log[255*255/MSE] is 66.71918212.

VI. Selective Watermarking against cutting/cropping Attacks

The above watermarking scheme can be used to watermark the image considering the portions of the image of block size 8X8 as illustrated in figure below.

Figure 11 Original Grayscale image of Lena

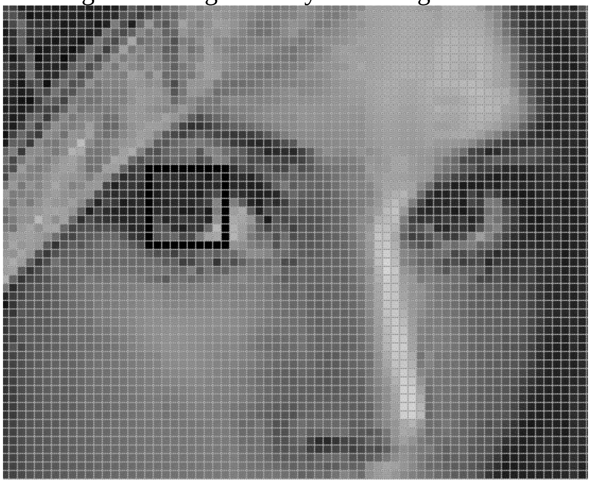

Figure 3.12 Watermarked Image of Lena (magnified 400 times). Watermarked in 8X8 portion of image shown in black square

Thus, any given image can be watermarked at some pre specified places so that watermark can be detected and can be extracted with high probability.

Moreover, the watermark can be embedded on such portions of the image (such as shown in the figure 3.10 above), such that the probability of cutting or cropping effects can be minimized. This technique can be applied to watermark several bit planes of the image based on the number of bit planes modified.

Watermarking specific portions of the image, selectively, with pre specified number of bit provide significant improvement over traditional DFT based watermarking, with improvements in the error probability of several orders of magnitude for most typical scenarios. PN sequences and Spread Spectrum is currently used by many watermarking schemes as the information embedding (or modulation) technology. Results provided in Chapter 4 clearly indicate that this correlation based image watermarking using m sequences provides a much robust watermarking as compared to existing technique of BPOF only watermarking technique which is mostly used as fragile watermark technique for image tempering detection. The proposed watermarking technique can be readily applied to practically any watermarking technique currently using PN sequences, taking immediate advantage of the gains.

VII. Effect of Image resizing

Image interpolation provides one of the most basic technique for image resizing. Interpolation is useful when one needs to reduce the file size or increase the amount of image data sent at one time. With the image size reduced, the image quality does not have to be significantly reduced as well.

Interpolation also provides a way through which images are enlarged. There are many different types of interpolation methods, each resulting in a different look to the final picture. Thus, it is best if the quality, or visible distinction for each pixel, is retained throughout the enlargement process. Thus, one cannot simply have a number of pixels directly represent a single original pixel. this type of interpolation is not sufficient for commercial use. Conspicuous blocks of single color will be visible, and depending on size of enlargement, the original image will be unrecognizable. Older methods of linear interpolation somewhat addressed this problem. By finding a mean pixel value between neighboring pixels, one can be able to produce an effect of blurred edges and smoothed details. Bilinear re-sampling uses the values from the four surrounding pixels, and new pixel values are calculated by weighting the averages of the four closest pixels based on distance. The new pixel value is determined by calculating a weighted average of the four closest pixels (2x2 array) based on distance. However, bilinear interpolation seems to work better for image reduction rather than image enlargement. Linear interpolation methods are simple and it has been found that nonlinear methods are superior. Some non-linear interpolation methods include Bi-Cubic, Soft Directional, and non-linear interpolation through extended permutation filters. Bicubic interpolation uses the nearest sixteen pixels (4x4 array) based on distance, which produces a much better effect than linear interpolation. Most high-end image manipulation and viewing programs today have some sort of interpolation process used when resizing an image. Such include ACDSee, Adobe Photoshop, IfranView, and even Internet Explorer. They implement many commercial interpolation methods like Lanczos Interpolation, standard Kneson, and pxlSmartScale.

(a) Nearest Neighbor Interpolation

The most basic form of interpolation is Nearest Neighbor Interpolation. In case of image enlargement, as the actual pixels are proportionally copied to their new locations, their position in relation to one another remains the same. Since the image is enlarged, filler pixels must be placed in between the actual pixels.

With the most basic nearest neighbor interpolation, the process is to just copy the exact same pixel values over to the filler pixel closest to the pixel. Consider the following figure 3.13.

22	28
16	30

o.	Pixel Values	FFT	Brightnes s Adjusted Pixels	FFT
1	24	361	27	409
2	15	-42.9789037926387-31.8348821071031i	18	-42.9789037926387-31.8348821071031i
3	26	18.857864376269+4.34314575050763i	29	18.857864376269+4.34314575050763i
4	19	-37.465218650608-10.7524330107325i	22	-37.465218650608-10.7524330107325i
5	35	47+6i	38	47+6i
6	16	-29.3632084741382-11.4975990147935i	19	-29.3632084741382-11.4975990147935i
7	30	47.1421356237309-15.6568542494924i	33	47.1421356237309-15.6568542494924i
8	21	-18.192669082615+59.4199518888359i	24	-18.192669082615+59.4199518888359i
9	56	53	59	53
10	19	-18.1926690826151-59.4199518888359i	22	-18.1926690826151-59.4199518888359i
11	9	47.142135623731+15.6568542494924i	12	47.142135623731+15.6568542494924i
12	28	-29.3632084741382+11.4975990147934i	31	-29.3632084741382+11.4975990147934i
13	12	47-6i	15	47-6i
14	24	-37.465218650608+10.7524330107325i	27	-37.465218650608+10.7524330107325i
15	15	18.8578643762691-4.3431457505076i	18	18.8578643762691-4.3431457505076i
16	12	-42.9789037926387+31.8348821071031i	15	-42.9789037926387+31.8348821071031i

Figure 13 A 4 Pixel Test Block

For nearest neighbor interpolation 4 times, the resultant is shown in fig 3.14.

22	22	28	28
22	22	28	28
16	16	30	30
16	16	30	30

Figure 14 Nearest Neighbor Interpolation Scaled Image (4 times scaled)

An image scaled as per nearest neighbor interpolation can be readily identified and techniques can be implemented in the detector for the detection of the watermark originally embedded in the watermarked non-scaled image.

(b) Linear Interpolation

A better algorithm than nearest neighbor that takes into account the gradual transition of pixel color values. By finding the means between two pixel values, the fillller pixel is better suited for overall image enhancement.

Consider again the same figure 3.4 repeated for ready reference.

22	28
16	30

With Filler pixels, the resultant is as shown in figure 15.

22	P	28	P
P	P	P	P
16	P	30	P
P	P	P	P

Figure 15 Linear Interpolation Scaled Image (4 times scaled)

For every P, one has to calculate the mean of the surrounding pixels. Eventually, one will be able to calculate the mean for every P, even those that were originally surrounded by all P. In the above example, the value of P comes out to be (22+28+16+30)/4.

It turns out that our proposed one dimensional FFT technique for watermark embedding fails when the image is scaled through linear interpolation. Thus, in case of linear watermarking, the proposed scheme serves as a technique for fragile watermarking.

(c)Image Brightness

Consider an image portion as shown in figure 16

24	15	26	19
35	16	30	21
56	19	9	28
12	24	15	12

Figure 16 Test Block 4X4 Pixels

Consider again the same image portion with adjusted brightness factor x, shown in figure 3.16

24+x	15+x	26+x	19+x
35+x	16+x	30+x	21+x
56+x	19+x	9+x	28+x
12+x	24+x	15+x	12+x

Figure 17 Test Block 4X4 Pixels (brightness increased with factor x)

Let x be 3. The FFT of original and brightness adjusted image is compared in the following table.

TABLE 2 PROPERTY OF FFT FOR UNIFROM INCREMENT/DECREMENT IN PIXEL VALUES (ADJUSTED BRIGHTNESS)

It turns out that our algorithm for image watermarking is robust against image brightness adjustment.

This can be understood with the following property of FFT.

$$f(x) \dashrightarrow F(u)$$
$$f(x+a) \dashrightarrow F(u)e^{-jau}$$

It is clear from the above table that frequency components remain unchanged when brightness is uniformly increased in pixels. Therefore, as per the proposed scheme of embedding, the modified bits in FFT magnitudes remain unchanged leading to the reliable detection of watermark even in brightness adjusted image.

VIII. Conclusion

The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image.

The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction, image compression and image watermarking. FFT based watermarking is robust against various types of attacks.

One obvious result can be derived from figure 4.9 in context of PSNR is this that it achieves a good value as the image size is reduced in case of grayscale image. This result follows from the fact that for images of size 32X32, the neighborhood of a specific pixel contains the intensity in some constrained range only, thus yielding smaller values for Mean Square Error and correspondingly, better PSNR.

REFERENCES

[1] Cox, M. Miller, and A. McKellips, "Watermarking as communications with side information," Proc. IEEE, vol. 87, pp. 1127–1141, July 2008.

[2] B. Chen and G. Wornell, "Achievable performance of digital watermarking systems," in Proc. Int. Conf. Multimedia Comput. Syst., Florence, Italy, June 207, pp. 13–18

[3] G.Depovre, et al., "Improved Watermark Detection Reliability Using Filtering Before Correlation" , IEEE International Conference on Image Processing, ICIP-98, Vol.1, pp.430-434.

[4] M. Miller, I. Cox, and J. Bloom, "Informed embedding: Exploiting image and detector information during watermark insertion," in Proc.IEEE Int. Conf. Image Process., 2002.J.

[5] Dittmann, A. Steinmetz, and R. Steinmetz, ‘‘Content-based digital signature for motion pictures authentication and content-fragile watermarking,’’ in IEEE Int. Conf. on Multimedia Computing and Systems, Vol. 2, pp. 209–213 2011.

[6] F. Prez-Gonzlez, F. Balado, and J. R. Hernndez, Performance Analysis of Existing and New Methods for Data Hiding with Known-Host Information in Additive Channels, IEEE Trans. Signal Processing, vol. 51(4), pp. 960{980, April, 2012.

[7] S.P.Mohanty, et al., ”A Dual Watermarking Technique for Images”, Proc. 7th ACM International Multimedia Conference, ACM-MM’99, Part 2, pp. 49-51, Orlando, USA, Oct. 2005.

[8] I.J.Cox et. al., ”Secure Spread Spectrum Watermarking of Images, Audio and Video”, Proc IEEE International Conf on Image Processing, ICIP-96, Vol.3, pp 243-246, http://www.neci.nj.nec.com/tr/neci tr 95 10.ps

[9] B. Javidi and E. Ahouzi, ‘‘Optical security system with Fourier plane encoding,’’ Appl. Opt. 37~26, 6247–6255 ~2008.

[10] Krishnan, N. Selvakumar, R.K. Rajapandian, S. Arul Mozhi, K. Nelson, Kennedy Babu, ”A Wavelet Transform Based Digital Image Watermarking and Authentication”, Annual IEEE Conference, in India, 15-17 Sept. 2006, pp: 1-6

[11] H. Malik, A. Khokhar, and R. Ansari, An Improved Detector/Decoder for Spread Spectrum based Watermarking using Independent Component Analysis, ACM Fifth Workshop on Digital Rights Management (DRM'05), November 7, 2005.

[12] J.R.Smith and B.O.Comiskey, ”Modulation and Information Hiding in Images”, Proc. of First International Workshop on Information Hiding, University of Canbridge, UK, May 30-June 1 1996, Lecture Notes in Comp. Sc., Vol.1174, Ross Anderson (Ed.).

[13] Abu-Errub, A., Al-Haj, A.,”Optimized DWT-based image watermarking”, First International Conference on Applications of Digital Information and Web Technologies, IEEE,2008, 4-6.

[14] D. Kirovski and H. Malvar, "Robust spread-spectrum audio watermarking," in Proc. Int. Conf. Acoust., Speech, Signal Process., Salt Lake City, UT, May 2001.

[15] Cichocki, S. Douglas, and S. Amari, Robust Techniques for Independent Component Analysis (ICA) with Noisy Data, Neurocomputing, Elsevier Science, vol. 22, pp. 113{129, 2001.

[16] Ahmad Nilchi1, Ayoub Taher,” A New Robust Digital Image Watermarking Technique based on the Discrete Cosine Transform and Neural Network”, pp:1 – 7, April 2008.

[17] Abu-Errub, A., Al-Haj, A.,”Optimized DWT-based image watermarking”, First International Conference on Applications of Digital Information and Web Technologies, IEEE,2008, 4-6. J.J.K. O’Ruanaidh, et al., ”Phase Watermarking on Digital Images”, Proc. IEEE International Conf. on Image Processing, ICIP-96, Vol.3, pp 239-242.

[18] R. Gribonval, L. Benaroya, E. Vincent, C. Fvotte, Proposals for Performance Measurement in Source Separation, 4th Int. Sym. Independent Component Analysis and Blind Source Separation (ICA'03), April, 2003.

[19] K.Hill, ”A Perspective: The Role of Identifiers in Managing and Protecting Intellectual Property in the Digital Age”, Proceedings of the IEEE, Vol.87, No.7, July 2009, pp.1228-1238.

[20] M.D.Swanson, et al., ”Multimedia data Embedding and Watermarking Technologies”, Proc. of the IEEE, Vol.86, No.6, June 2006, pp.1064-1087.

[21] S.Craver, ”On Public-Key Steganography in the Presence of an Active Warden”, Proc. of the 2nd International Workshop on Information Hiding, Portland, Oregon, USA, 15-17 Apr 1998, Lecture notes in Comp Sc, Vol.1525, Springer-Verlag.

[22] Z. Tirkel, C. F. Osborne, and R. G. van Schyndel, "Image watermarking—A spread spectrum application," in Proc. IEEE 4th Int. Symp. Spread Spectrum Techn. Applicat., Mainz, Germany, 2006, pp. 785–789

[23] J. Fridrich and M. Goljan, ‘‘Images with self-correcting capabilities,’’ in Proc. Int. Conf. on Image Processing, Vol. 3, pp. 792–796 ~2005.

