International Journal of Computer Theory and Engineering, Vol. 1, No. 4, October, 2009
1793-8201

Blind Image Fidelity Assessment Using the Histogram

M. I. Khalil

***Abstract*—An image fidelity assessment and tamper detection using two histogram components of the color image is presented in this paper. The histograms of the red and green channels of the to-be-protected image are embedded in the green and blue channels respectively. The embedded histogram gets destroyed whenever any sort of modification is made to the content of the image yielding to mismatch during the detection process. In addition to the capability of detecting geometric transformation, removal of original objects and addition of foreign objects, the proposed algorithm is also capable of detecting cropping. Experimental results show that embedding the histogram data within the image in this manner does not deteriorate the quality of the original image.**

***Index Terms*—Fidelity Assessment, Histogram, Image, Spatial Domain, Watermarking.**

I. INTRODUCTION

Tampering proof is one of the important applications of image watermarking which is a technique of embedding signal or logo into images. Any tampering operations onto a watermarked image can be detected by verifying the watermarked image in a certain way. It is generally preferred to design a verification process that proceeds without referencing the original logo or the non-tampered image. Such watermarking techniques may be used for medical images, news images, etc. It is intended to develop and implement a multimedia system which includes Holly digital media, like audio, video, images, and other multimedia documents. It is urgently desirable to protect this digital media against non-ethic attempts to change the contents or replace it with improper objects. The function of this system is not to prevent these non-ethic attempts but to recognize whatever the to-be-used items are the original ones or not. Digital watermarking is mainly used for copy-protection and copyright-protection [1,2]. The copy protection attempts to find ways, which limits the access to copyrighted material and/or inhibit the copy process itself. On contrast, the copyright protection inserts copyright information into the digital object without the loss of quality [3-5]. Watermarking technology can also be used to guarantee authenticity and can be applied as proof that the content has not been altered since insertion. The watermark is often designed in such a way that any alteration either destroys the watermark or creates a mismatch between the content and the watermark, which can easily be detected. This paper is dedicated to deal with the digital images included in this system. The system should be responsible of watermarking the included images before distributing the package. In the client side the program checks every image, before displaying it, to ensure that the included media have not been tampered with and to prove their true origin. Achieving this purpose, several watermarking techniques have been surveyed to find one that matches with the system requirements. Schneider et al. [6] proposed a scheme based on public key encrypted image block histogram for the purpose of authentication. The Euclidean distance between histogram of each block of the original image and the histogram of each block of the watermarked image is calculated. An authenticity measure is subsequently calculated by summing all the Euclidean distances over the entire image and compared against a pre-specified threshold for authentication. The major drawback of this scheme is twofold: First, it is not secure enough because modifying an image without altering its histogram is trivial. Secondly, a large database of the public key encrypted histogram is required. Chee Sun Won [7] proposed a scheme for image fidelity assessment using the edge histogram descriptor (EHD) of MPEG-7. In this scheme neither additional data nor fragile watermarking is needed, and there no need to access the original image as a reference. Only the EHDs of the original image and the received image are required. The small color and geometrical modifications cannot be detected using this scheme.

II. THE PROPOSED ALGORITHM

As a matter of fact, the histogram bins can be considered as the true host feature set and it can used to analyze and classify images. Histograms have been found experimentally to have low sensitivity to certain types of image morphisms. Coltuc, et al. considered directly the image histograms as watermarks [8]. Histogram modification has been employed in several image watermarking techniques [9-11]. The histogram, in the suggested method, is embedded as a watermark inside the image stream itself instead of saving it in a separate stream. The embedded watermark as well as the image itself will be destroyed when the attacks are mounted on the image. The distortion of the image and consequently its histogram leads to mismatch when compared with the embedded histogram. This situation makes it impossible for any trivial attack to occur without detection. The watermarking algorithm can be represented in the framework shown in the block diagram of Fig.1. It is known that the histogram of a color image can be

Manuscript received April 27, 2009. M. I. Khalil. Egyptian Atomic Energy Authorit, Nuclear Research Center: Phone: 966-564680744.

solved into three components h_R, h_G, h_B corresponding to the three main color primaries R, G, B respectively. Let us study the possibility of embedding two components only of the three components h_R, h_G, h_B. The histogram of the red channel of the host image is computed and embedded into the least significant bits of the green channel of the image. The same thing is done with the histogram of the green channel. Consequently, the histogram of the green channel is computed and then embedded into the least significant bits of the blue channel of the image. Once this process is applied to on an image, it is impossible to get it back to the original status. Algorithm-0 and Table-1 explains the main features of the proposed algorithm.

Algorithm-0 : Watermarking
Input: 24 bit color image *with m x n pixels*, each color plane may be treated as a monochrome image and accordingly, the original image I consists of I_r , I_g and I_b components and is defined as follows:
$I = \{p(i,j)\mid 0 \le i \le m-1,\ 0 \le j \le n-1,\ 0 \le p(i,j) \le 255\}$
Output: Watermarked Image $\tilde{I}$ consists of (I_r , $\tilde{I}_g$ and $\tilde{I}_b$ components)
Step 0: Extract histogram $h_r = F(I_r)$
Step 1: Embed h_r **into** I_g yielding $\tilde{I}_g$
The embedding process will be explained later,
Step 2: Extract histogram $\tilde{h}_g = F(\tilde{I}_g)$
Step 3: Embed $\tilde{h}_g$ **into** I_b yielding $\tilde{I}_b$

Algorithm-1 : Extracting Histogram
Input: Image channel (I_r or $\tilde{I}_g$)
Output: Histogram (h_r or $\tilde{h}_g$)
Step 0: Extract histogram $h_r = F(I_r)$
The basic algorithm for creating a histogram is very simple:

```
int h[256];
for (i=0;i<m;i++)      {
    for(j=0;j<n;j++)      {
        retrieve pixel[i,j]
        // compute h index c for pixel
        c=color value of the pixel
        h[c]++;
        }
    }
```

Table I WATERMARKING ALGORITHM IN STEPS

Initial status	Step-0		Step-1		Step-2		Step-3	
Image components	Action	Result	Action	Result	Action	Result	Action	Result
I_r	Compute histogram	h_r						I_r, h_r
I_g			Embed h_r into I_g	$\tilde{I}_g$	Compute histogram	$\tilde{h}_g$		$\tilde{I}_g, \tilde{h}_g$
I_b							Embed $\tilde{h}_g$ into I_b	$\tilde{I}_b, \tilde{h}_b$

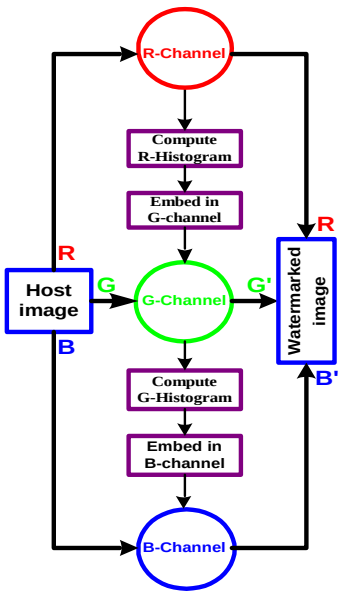

Fig.1 Watermarking algorithm

A. Computing Histogram

The color histogram is defined by: $h_A(a) = N.\text{Prob}(A=a)$ Where A represents a color channel (R, G or B) and n is the number of pixels in the image. Computationally, the color histogram is formed by discretizing the colors within an image (m x n) and counting the number of pixels of each color. Algoritm-1 contains a fragment pseudo code to extract the histogram for a color component of an image.

B. Watermark Embedding

Watermark embedding is performed by**:** The histogram of the red channel of the host image is computed and saved in an integer array which in turn is transformed into a bit string b_0 b1::: bn. The content of this string is then embedded bit by bit into the least significant bits of the green channel of the image. The same thing is done with the histogram of the green channel. The histogram of the green channel could not be computed before saving the histogram of the red channel because its values will change as a result of embedding the histogram of the red channel within it. Consequently, the histogram of the green channel is computed and saved in an integer array which in turn is transformed into a bit string b_0 b_1 ::: b_n before embedding it into the least significant bits of the blue channel of the image. It is not possible to save the histogram of the blue channel in the red channel because the histogram of the red channel has computed and saved in the green channel and any alteration will cause distortion to integrity of the algorithm and mismatch between the histogram of the red channel that is previously saved in the green channel and the newly computed one. The embedding process is key dependent in order to make it impossible, without knowledge of the procedure and the secret key, to remove the watermark or to make it illegible.

Algorithm-2 : Embedding histogram
Input: Image channel (I_g and h_r) *or* (I_b and $\tilde{h}_g$)
Output: Watermarked channel ($\tilde{I}_g$ or $\tilde{I}_b$)
Step 0: Transform the histogram into a bit string b_0 b_1 ::: b_n
Step 1: The content of this string is then embedded bit by bit into the least significant bits of the I_g or I_b channels of the image. This can be achieved simply by sequentially replacing the least significant bit by the corresponding bit of the histogram binary string.

C. *Fidelity Assessment*

Watermark extraction and image authentication is preformed as shown in Fig. 2. After extracting the watermark, the result of authentication is performed by comparing the original watermark and the extracted watermark. As blind, fidelity assessment, the detector D is a two-argument function accepting as input the to-be-checked image I, retrieved histogram h_s. As an output D decides whether I has been corrupted or not, that is: **$D(I, h_s)$ = yes/no**
The detection algorithm does not require the original image but the position and the order of embedding histogram streams must be known prior to performing the detection process. The detection process, as shown in Fig.2, begins by computing the histogram of the Red channel. The stored histogram (h_r) of the Red channel of the original image is retrieved from the Green channel. Both of the histograms of the Red channel ($h_{r\text{-}omputed}$, h_r) are compared and if they are not coincident then it means that the image is corrupted. Then, the histogram of the Green channel can be computed and compared with that retrieved from the blue channel ($\tilde{h}_g$) , and if they are not coincident then the image is corrupted.

III. Experimental Results

In this section, some experimental results are demonstrated to show the effectiveness of the proposed watermarking schemes. Both the histogram embedding and fidelity assessment algorithms have been implemented in visual C#. The histogram embedding algorithm has been applied to number of images, then some of these images exposed to some color and geometrical modifications: content modification, and image rotation. The process of fidelity assessment and tamper detection is then applied to these images. Result with no modification is shown in Fig. 3. When comparing the histogram components in the left side of the figure which correspond to the original image with that in the middle which correspond to the watermarked image, we can note a shadow around the Green and Blue histogram components of the watermarked image. This shadow is due to embedding data in the Green and Blue channels through the watermarking process. The embedded histogram components are then extracted from the Green and Blue channels of the watermarked image and displayed in the right side of the figure. For the watermarked image, the computed histogram components and that extracted during the detection process are identical as long as it is not attacked. Result to content modification is shown in Fig. 4; another image is watermarked and then corrupted by swapping two flags within the image. The Red and Green histogram components are computed (the left side of the figure). The corresponding embedded histograms are retrieved from the Green and Blue channels respectively (the right side of the figure). It is easily noticeable that the two histograms are completely different due to the corruption applied to the watermarked image. Result to image rotation is shown in Fig. 5; another image is watermarked and then rotated counterclockwise 90^0. Result to JPEG compression is shown in Fig. 6. As general, any modification that led to changes in the histogram of the images has been detected and accordingly classified as image corruption. During the

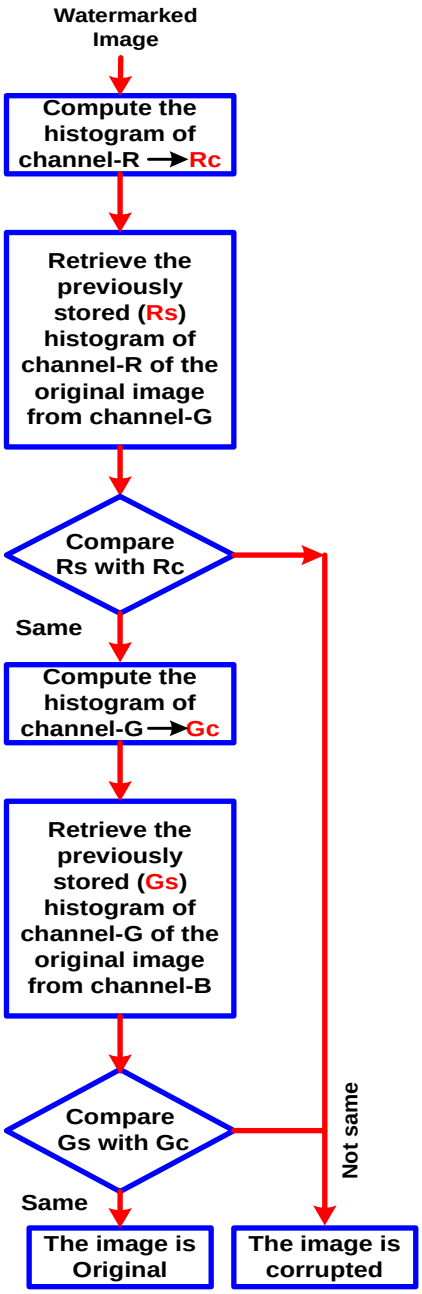

Fig.2 Tampering proof / fidelity assessment algorithm

previous tests, it was clear that the proposed embedding algorithm does not distort host image much because they looks almost identical. To measure the distortion incorporated by the watermarking algorithm, MSE (Medium Square Error) and PSNR (peak signal-to-noise ratio) have been used for color images with color components R,G and B.

It is given by:

$$PSNR = 10\log_{10}\left\{\frac{255^2}{\frac{MSE(R)+MSE(G)+MSE(B)}{3}}\right\}$$

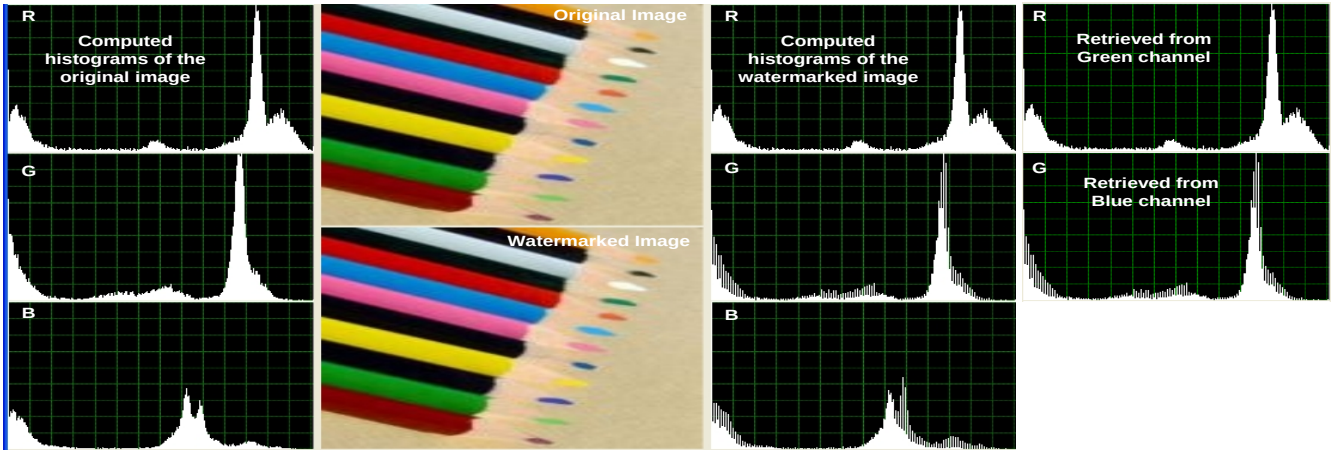

Fig. 3 An example of watermarking and detection processes.

Fig.4 The watermarked imaged is corrupted by swapping of two flags.

Fig. 5 Computed histogram of a watermarked image and the extracted one after rotating the same image.

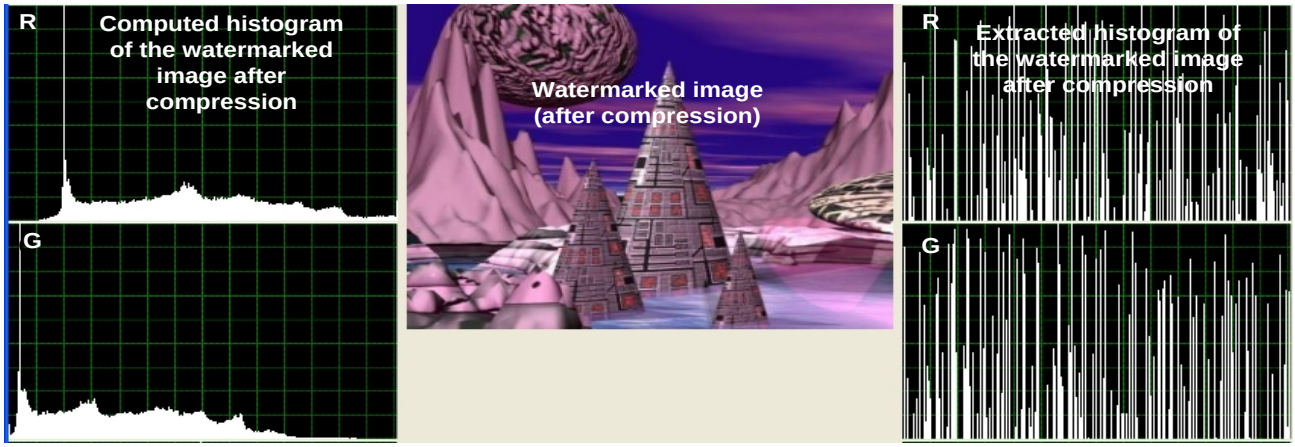

Fig. 6 Computed histogram of a watermarked image and the extracted one after compressing the same image.

Here ***MSE*** represents the mean square error:

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (X_{ij} - X'_{ij})^2$$

Where X_{ij} and X'_{ij} represent the original and secured images respectively. A large value for ***PSNR*** means less difference between the original image and the secured (watermarked) one. In general, it is difficult to recognize the difference between the original image and watermarked one in vision if the PSNR value is greater than 30 dB. The computed lower bound of the PSNR for the previous test cases is 53 dB. The PSNR, using this technique, increases as the size of the image increases. This can be referred to the fact that the embedded histogram replaces fixed number of the least significant bits of the watermarked image (2 histogram components x 255 counts for each histogram x 32 bits for representing each count). In the worst case, all modified pixels will be added or subtracted by 1. The mean squared error of this case is 0.5 and accordingly:

MSE(R)=0 , MSE(G)=255 x 32 x 0.5 / (m x n) ,
MSE(B)=255 x 32 x 0.5 / (m x n)

$$PSNR = 10 \log_{10} \left\{ \frac{255^2}{\frac{0 + 255x32x0.5 + 255x32x0.5}{3\,mxn}} \right\} = 10 \log_{10} \left\{ \frac{3x255xmxn}{32} \right\}$$
$$= 10 \log_{10} \{24\,m\,x\,n\}$$

Accordingly, the relation can be simplified to take the following form: ***PSNR = 10 Log$_{10}$ {24 m x n}***,
This relation has been experimentally demonstrated in Fig.7.

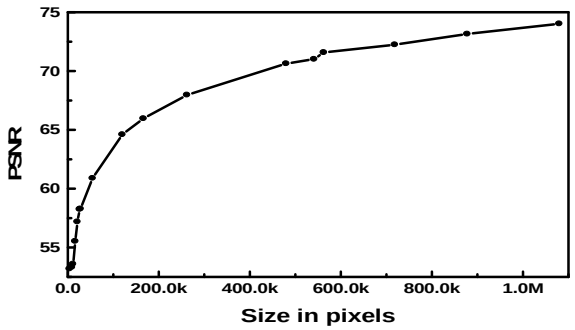

Fig. 7 :PSNR versus image size (pixels)

IV. Conclusion

A spatial domain blind watermarking algorithm has been proposed in this paper. The histograms of the red and green channels of the to-be-protected image are embedded in the green and blue channels respectively. The embedded histograms are to be destroyed when even trivial attack is mounted on the image yielding to mismatch during the detection process. The algorithm is shown to be very sensitive to any intentional and unintentional attacks. Moreover, the detection process is conducted without referencing the original image. Experimental results have demonstrated reliability and validity of the algorithm.

References

[1] I. J. Cox, M.L. Miller, and J.A. Bloon, "Watermarking Applications and their Properties", in the proceeding of Int. Conf. on Information Technology 2000, Las Vegas, 2000.

[2] Ping Wah Wong and Nasir Memon, "Secret and public key watermarking schemes for image Authentication and ownership verification", IEEE transactions on image processing, 10(10),1593-1600, (2001).

[3] Jeng-Shyang Pan, Hsiang-Cheh Huang, L. C. Jain, *Intelligent Watermarking Techniques (Innovative Intelligence)*, World Scientific Publishing Company, (2004).

[4] Jeng-Shyang Pan, Hsiang-Cheh Huang, Lakhmi C. Jain, Wai-Chi Fang, *Intelligent Multimedia Data Hiding: New Directions*, Springer, (2007).

[5] Mauro Barni and Franco Bartolini, *Watermarking Systems Enginnering*, Marcel Dekker, Inc., (2004).

[6] M. Schneider and S. F. Chang, "A robust content based digital signature for image authentication", *IEEE Intl. Conf. On Image Processing*, *3*,227-230,(1996).

[7] Chee Sun Won, "Image Fidelity Assessment Using the Edge Histogram Descriptor of MPEG-7", *ETRI Journal*, 29(5),(2007).

[8] Coltuc, D. and Bolon, P., "Robust watermarking by histogram specification", *International Conference on Image Processing*, 2, 236 – 239, (1999).

[9] Chee Sun Won, "Image Fidelity Assessment using the Edge Histogram Descriptor of MPEG-7", *ETRI Journal*, 29(5),(2007).

[10] Yang, B, Schmucker, M, Busch, C, Sun, S, „Reversible image watermarking by histogram modification for integer DCT coefficients", *IEEE Workshop on Multimedia Signal Processing 2004. Proceedings.,New York, NY: IEEE, 2004*,143-146

[11] Roy, S.; Chang, E.-C, "Watermarking color histograms, Image Processing", *ICIP '04. 2004 International Conference*, 24-27 Oct. 2004, 4, 2191 - 2194 (2004).

M. I. Khalil. received his B.Sc degree in Computer and Automatic Control Engineering from Ain Shams University, Cairo, Egypt, in 1983, M.Sc degree in Computer Engineering from Tanta University,Tanta, Egypt, in 2003 and Ph.D degree in Computer System Engineering from Benha University, Cairo,

Egypt, in 2005. He is currently working as Assistant Professor in Department of Computers at the College of Science, Princess Noura Bent Abdulrahman University, Riyadh, KSA. He has 15 years of previous experience at the Reactor physics Department, Nuclear Research Center, Cairo, Egypt in the field of Data Acquisition and Interface Design. His area of interest includes image processing and Digital Signal processing.