

# Hiding Watermark in Image During Lossless Wavelet Compression With Adaptive 2D Decomposition

**Dr. Amir S. Al-Malah**

Ph.D Computer Science  
Al\_ Mustansyria University  
College of Science  
Computer Science Dept.

**Haider Kadhim Homood**

Msc Computer Science  
Al\_ Mustansyria University  
College of Science  
Computer Science Dept.

**Jolan Rokan N.**

Msc Computer Science  
El\_Imam Ja'afar al\_sadeq  
University  
IT College

## Abstract

Digital watermarking, a technique being used in the copyright protection recently, can be applied to images for the purpose of authentication. The only difference between these two uses of digital watermarking is that the watermarks embedded to an image for the purpose of copyright protection should be very robust to resist any manipulation, while they should be sensitive to malicious manipulations and robust to compressions for the purpose of image authentication.

In this paper a new robust image authentication method, which provides an approach to produce such kind of digital watermarks techniques using adaptive compression related applications is presented. The adaptive 2D decomposition selects 2D wavelet functions based on the compression of the coefficients.

## 1. Introduction

Image authentication techniques use either external signatures or embedded watermarks to verify the originality of an image. An external authentication signature generated from the original

image is usually an encrypted form of some kind of its hash values. The authentication process of a pending image depends on whether the hash

values decrypted from the signature match its hash values [1] or whether the constraints set by the hash values of the original image can be satisfied by this pending image [2]. The second form of authentication method is based on the embedded watermarks. This technique embeds symbols to an image. Because the embedded symbol (watermark) will be modified only if the image is manipulated, the authentication process of this method is based on the detection of the watermark. A robust watermark should be still detectable even if the image is lossy compressed, and is sensitive to malicious manipulations, such as cloning and replacing. There is a common drawback to any watermark technique, the possible quality descending of the image in the process of symbol embedding. However, the benefit of this technique is its efficiency, while all protection of images is within the images themselves such that no additional signature is needed for authentication. The watermark-based authentication is shown in Fig. 1.

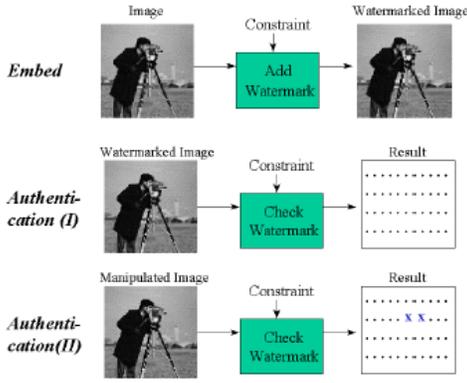


Fig. 1: Watermark-based Image Authentication Method

## 2. Wavelet Principal

Wavelets and wavelet transformations have a wide variety of different applications in computer graphics including radiosity, multiresolution painting, curve design, mesh optimization, volume visualization, image searching, animation control [3], BRDF representation [2], and, one of the first applications in computer graphics, image compression [3], wavelets and wavelet transforms can become as important and ubiquitous in computer graphics as spline based techniques are now.

## 3. Orthonormal Wavelets in 1D

This short introduction deals only with a subset of wavelets. A more detailed overview of the different kinds of wavelet transforms including continuous wavelet transform, frames, and biorthogonal wavelets can be found in [4] and [5].

The orthonormal wavelet transform is based on two functions  $\psi(x)$  and  $\phi(x)$ , which have the properties:

$$\int \phi(x)dx = 1; \quad \int \psi(x)dx = 0 \quad (1)$$

These functions with their translations and dilatations  $\psi_{j,k}(x)$  and  $\phi_{j,k}(x)$  build an orthonormal basis and therefore any

function in  $L^2(\mathbb{R})$  can be reconstructed with these basis functions.  $\phi(x)$  is called scaling- or smooth-function, and  $\psi(x)$  wavelet or detail function.  $\psi_{j,k}(x)$  and  $\phi_{j,k}(x)$  can be constructed from their mother functions  $\psi(x)$  and  $\phi(x)$  in the following manner:

$$\phi_{j,k}(x) = \sqrt{2^j} \phi(2^j x - k), \quad \psi_{j,k}(x) = \sqrt{2^j} \psi(2^j x - k), \quad j, k \in \mathbb{Z} \quad (2)$$

$\{\phi_{j,k} \mid k \in \mathbb{Z}\}$  form an orthonormal basis of functions in vector space  $V_j$ . These vector spaces are nested, that is,  $V_0 \subset V_1 \subset V_2 \subset V_3 \subset \dots$ . Given a function  $f(x)$  over  $[0,1]$ , this function can be approximated in  $V_j$  with  $2^j$  scaling coefficients  $s_k^j$ :

$$f^j(x) = \sum_{k=1}^{2^j-1} s_k^j \phi_{j,k}(x) \quad \text{with } s_k^j = \langle f(x), \phi_{j,k}(x) \rangle \quad (3)$$

Also the detail functions  $\{\psi_{j,k} \mid k \in \mathbb{Z}\}$  form an orthonormal basis of functions in the detail vector space  $W_j$ , which is the orthogonal complement of  $V_j$  in  $V_{j+1}$ .  $W_j$  can be thought of as containing the detail in  $V_{j+1}$ , which can not be represented in  $V_j$ . The vector space  $V_{j+1}$  can be decomposed in the following manner:

$$V_{j+1} = V_j \oplus W_j = V_{j-1} \oplus W_{j-1} \oplus W_j = \dots = V_0 \oplus W_0 \oplus W_1 \oplus \dots \oplus W_j \quad (4)$$

Let  $d_k^j$  be the detail coefficients, given through:

$$d_k^j = \langle f(x), \psi_{j,k}(x) \rangle = \int f(x) \psi_{j,k}(x) dx, \quad (5)$$

then  $f^j(x)$  can be calculated from the detail coefficients  $\{d_k^i \mid i < j\}$  and the scaling coefficient  $s_{00}$  as follows:

$$f^j(x) = f^{j-1}(x) + \sum_{k=0}^{2^{j-1}-1} d_k^{j-1} \psi_{j-1,k}(x) = s_{00}^0 \phi_{0,0}(x) + \sum_{m=0}^{j-1} \sum_{k=0}^{2^m-1} d_k^m \psi_{m,k}(x) \quad (6)$$

The calculation of the coefficients  $\{s_{00}, d_k^i \mid 0 \leq i < j; 0 \leq k < 2^i\}$  from the scaling coefficients  $\{s_k^j \mid 0 \leq k < 2^j\}$  is called wavelet transformation. The fast wavelet transformation uses a pyramid scheme with two subband filters, the

smoothing or scaling filter ( $h_m$ ), and the detail or wavelet filter ( $g_m$ ). In one transformation step the  $2^i$  scaling coefficients  $s_{ki}$  are replaced by  $2^{i-1}$  scaling coefficients  $s_{k_{i-1}}$  and  $2^{i-1}$  detail coefficients  $d_{k_{i-1}}$ :

$$s_k^{i-1} = \sum_m h_{2k-m} s_m^i; \quad d_k^{i-1} = \sum_m g_{2k-m} s_m^i \quad (7)$$

This step is repeated on the remaining scaling coefficients, until  $s_{00}$  is computed. The reconstruction step can be performed using the adjoint filtering operation:

$$s_k^i = \sum_m h_{k-2m} s_m^{i-1} + g_{k-2m} d_m^{i-1} \quad (8)$$

#### 4. Wavelets in Higher Dimensions

The definitions in above section deal with wavelets in 1D space, but for image compression wavelet transformations in 2D are needed. One way to extend the formulas to 2D is to use a dilatation matrix  $D$  in (2) as described in [6], instead of a simple dilatation factor, e.g. the quincunx scheme uses the dilatation matrix  $D = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$

More popular methods extend the one dimensional wavelets to higher dimensions with tensor products of 1D wavelets and scaling functions, i.e. the rectangular and the square wavelet basis functions.

#### 5. The Rectangular Decomposition

The rectangular or standard wavelet basis functions are generated through the Cartesian product of the 1D wavelet basis functions in every dimension. In the 2D case, the rectangular wavelet basis functions are:

$$\left. \begin{aligned} \phi(x,y) &= \phi_{0,0}(x)\phi_{0,0}(y) && \dots 2D \text{ scaling function} \\ \text{R1 } \psi_{j,k}^1(x,y) &= \psi_{j,k}(x)\phi_{0,0}(y) \\ \text{R2 } \psi_{j,k}^2(x,y) &= \phi_{0,0}(x)\psi_{j,k}(y) \\ \text{R3 } \psi_{j,m,k,n}^3(x,y) &= \psi_{j,k}(x)\psi_{m,n}(y) \end{aligned} \right\} \begin{aligned} & \dots 2D \text{ wavelet functions} \\ & 0 \leq j, m < L; \\ & 0 \leq k < 2^j; 0 \leq n < 2^m \end{aligned}$$

The fast wavelet transformation with the rectangular basis wavelets, also known as rectangular decomposition, is computed by successively applying the 1D wavelet transformation to the data in every dimension. In the 2D case, all the rows are transformed first, then a 1D wavelet transformation is applied on all columns of the intermediate result.

Fig.2a illustrates the rectangular decomposition. The wavelet coefficients of the 1D transformation steps are stored in the right (row transform) or lower (column transform) part, the scaling coefficients in the left or upper part, respectively.

#### 6 The Square Decomposition

The square or nonstandard wavelet basis functions are also generated through Cartesian product of 1D wavelets and 1D scaling functions. In contrast to the rectangular basis functions, the square basis functions always use tensor products of wavelet and/or scaling functions of the same resolution level. In the 2D case, the square wavelet basis functions are:

$$\left. \begin{aligned} \phi(x,y) &= \phi_{0,0}(x)\phi_{0,0}(y) && \dots 2D \text{ scaling function} \\ \text{S1 } \psi_{k,m}^1(x,y) &= \psi_{j,k}(x)\phi_{j,m}(y) \\ \text{S2 } \psi_{k,m}^2(x,y) &= \phi_{j,k}(x)\psi_{j,m}(y) \\ \text{S3 } \psi_{k,m}^3(x,y) &= \psi_{j,k}(x)\psi_{j,m}(y) \end{aligned} \right\} \begin{aligned} & \dots 2D \text{ wavelet functions} \\ & 0 \leq j < L; 0 \leq k, m < 2^j \end{aligned}$$

The square wavelet decomposition can be computed with a similar technique as the rectangular decomposition: A 1D wavelet transformation step is applied in every dimension. This generates  $(2^{\dim} - 1)$  subbands with wavelet coefficients and one subband with scaling coefficients. This transformation scheme is applied recursively on the scaling coefficients until the lowest level is reached (Fig. 2b). The square decomposition is slightly more efficient to compute than the

rectangular decomposition: For an  $m \times m$  image only  $(8/3)(m^2-1)$  assignments are needed, compared to  $4(m^2-m)$  in the rectangular decomposition. Also the compression ratios are usually better for the square decomposition, because the support of the wavelet functions are square and support width of the wavelet basis functions are lower or equal than their counterparts in the rectangular decomposition and therefore they exploit more locality.

### 8. Adaptive 2D Wavelet Decomposition

Let us take a closer look at the difference of the square and the rectangular decomposition in the 2D case: The first obvious similarity of the two schemes is the use of the same scaling function  $\phi(x,y)$ , whose coefficient is stored in the upper left corner of the transformed images. In one decomposition step of the square decomposition three wavelet subbands and one scaling subband are generated. The wavelet subbands are not altered in the following decomposition steps. The first wavelet subband in the upper right part of the transformed image consists of the coefficients of the wavelet functions  $h\psi$ , the second subband in the lower left part has the coefficients of  $v\psi$ , and the third subband in the lower right part holds the coefficients of  $d\psi$ . All the wavelet functions in  $d\psi$  are also contained in  $R3\psi$  of the rectangular decomposition, therefore the coefficients in this part of the transformed image are the same for both decompositions. The upper right part of the square transformed image contains coefficients of the wavelet functions  $h\psi^L(x,y) = \psi^L(x)\phi^L(y)$ , where  $L$  is the maximum resolution level. The corresponding rectangular decomposition holds the

coefficients of the wavelet functions  $\{R1\psi^L(x,y); R3\psi^{L,i}(x,y) \mid 0 \leq i < L\}$  ( as shown in figure 3). It can be seen from the definition of these wavelet functions in (9)(10), that this part of the transformed image in the rectangular decomposition can be generated from the square decomposition with a 1D wavelet transformation within every column of this part. In analogy, the lower left part of the rectangular decomposition can be generated from the square decomposition with a 1D wavelet transformation within every row. This relation between the square decomposition and the rectangular decomposition remains also in the following decomposition steps of the square decomposition with the resolution level  $L$  reduced by 1 from the above step. This observation leads to an alternative construction scheme for the rectangular decomposition:

- apply a square decomposition step
  - for every column in the upper right part: apply a 1D wavelet transform in the y-dimension
  - for every row in the lower left part: apply a 1D wavelet transform in the x-dimension
  - apply this scheme recursively on the upper left part of the transformed image
- The 1D wavelet transformation consists of an iteration of transformation steps. The idea of the adaptive 2D decomposition is to replace the 1D transformations in the alternative construction scheme of the rectangular decomposition with an optimal number of transformation steps in respect to the compression rate of the coefficients. The pseudo-code of the adaptive 2D decomposition can be written as:
- apply a square decomposition step
  - for every column in the upper right part:

apply all 1D wavelet decomposition steps in the y-dimension calculate the compression rates for all steps select the number of steps with optimal compression rate

- for every row in the lower left part:

apply all 1D wavelet decomposition steps in the x-dimension calculate the compression rates for all steps select the number of steps with optimal compression rate

- apply this scheme recursively on the upper left part of the transformed image

Note that this adaptive 2D decomposition also includes the rectangular and the square decomposition: If the square decomposition has the best compression of the coefficients, the adaptive 2D decomposition selects the square decomposition wavelet functions. The same is true for the rectangular decomposition. In the general case, the adaptive 2D decomposition selects some wavelet functions from the square decomposition, some from the rectangular decomposition and some "between" the square and the rectangular decomposition.

There are only  $4(m^2-m)$  coefficient assignments needed to do the transformation for an  $m \times m$  image, the same number as for the rectangular decomposition. The inverse transform even needs less coefficient assignments, since the number of transformation steps is usually lower than the maximum.

There is a slight overhead for storing the number of 1D transformation steps: For an  $m \times n$  image less than  $m \cdot (\log_2(n)-1) + n \cdot (\log_2(m)-1)$  bits are needed for storing the adaptive decomposition. For example, an  $1024 \times 768$  image with 8 bit graylevels needs 768 kBytes for the uncompressed image and about 16

kBytes for the additional data, only about 2% of the original.

## 9. Wavelets for Lossless Image Compression

Even though there are many papers about wavelet based image compressions, only few deal with the lossless case [9]. Lossless wavelet image compressions use a 2D wavelet transform to improve the compression rate of conventional compression algorithms like Huffman or arithmetic coding [17]. Since the pixels have to be reconstructed exactly, some special properties for the wavelet transformations are required.

Let us first consider graylevel images: There is one color channel with a finite number of possible values, it usually has 8 bit depth or 256 shades of gray. For lossless compression it must be guaranteed, that the inverse transformation of the transformation does not change the pixels. This can be achieved for all compact wavelets, if the precision of the transformed image is high enough. But for high compression of the coefficients, the needed precision should be as low as possible. Bekaert et al. [7] used unnormalized Haar wavelets for lossless image compression. Since the least significant bit of the scaling coefficient is redundant, all wavelet coefficients can be stored in 9 bit, in the case of a cartesian product of a 1D scaling function with a 1D wavelet.

## 10. Embedding Watermark

The proposed procedure of watermark embedding is shown in Fig. 3. For each image, we have to designate a wavelet basis and a Pseudo Noise pattern. The Pseudo Noise pattern is used as a label for embedding and the wavelet basis is for protection. At the first step, the image is decomposed into four subbands, LL, LH, HL, and HH,

using 2D adaptive decomposition. Then, we can use the Pseudo Noise pattern to generate a substitution for HH subband and discard the original one. For example, a Pseudo Noise pattern of 16 x 16 pixels that are spatially repeated in the vertical and horizontal directions can be served as a HH subband substitution. At the last step, the watermarked image is obtained after applying the inverse wavelet transform.

The wavelet basis and the Pseudo Noise pattern are designated by the authentication system. They are secret to the attacker. An authenticator can use a serial number that is included in the watermarked image to decide which wavelet basis should be used for the Wavelet Transform and what kinds of Pseudo Noise pattern should be in the HH subband. The embedding process changes the HH subband of the image. Because HH subband information is usually insensible to people, this change will not introduce too much quality degradation to the image. Also, due to using the lossless wavelet compression the watermark in HH subband will be not effected.

### 11. Authentication Process

The authentication process is based on the detection of the existence of embedded Pseudo Noise pattern. At the first step of the authenticator, the HH subband of a pending watermarked image is extracted. It is convolved with the Pseudo Noise pattern. Because the autocorrelation of a Pseudo Noise pattern has a pulse in the origin and zeroes everywhere, if the image is not manipulated, the convolution result should be like a dot matrix. It is shown in Fig. 4(a).

Authentication of the image depends on whether the output of authenticator is a dot matrix. If the image is manipulated

by some filtering methods such as blurring and edge enhancing, the Pseudo Noise result will look like a dot matrix as long as the change is not drastic. If the image is compressed by JPEG lossy compression, the dot matrix should be still there. However, if some attacker manipulated it by replacing pixel values, the embedded Pseudo Noise pattern in that manipulated area should have been changed. Therefore, the convolution result of the corresponding area will be dispersed instead of a dot.

### 12. Experimental Results

Three experiments are shown in Fig. 4. Fig. 4(a) is the result of the watermarked original image, which shows that a uniformly distributed dot matrix can be obtained in this case. In Fig. 4(b), the watermarked image is manipulated at the ribbon area. Those area has been removed and replaced by the nearby background pixels. From the authentication result of it, we can observe the convolution values in the corresponding area have been dispersed. Therefore, we can detect the manipulation.

A JPEG compressed image is shown in Fig. 4(c). The compression ratio is 3: 1 in this case. We can observe that the authentication result is still like a dot matrix. Therefore, this image is also authentic in this case. Some other experiments showed that, if the image is compressed with a larger JPEG compression ratio, the watermarks degrade and the dot matrix is no longer to be distributed uniformly. It becomes harder to authenticate an intensively compressed watermarked image.

### 13. Discussion and Further Work

The watermark-based method is an efficient way for image authentication. The robust method that we proposed in this paper has proved its effectiveness by some experiments. Some further research issues including the enhancement of watermark robustness in the high compression ratio and the application of robust watermark-based authentication on image/video are being conducted.

The new adaptive 2D decomposition scheme offers better compression rates than the square and the rectangular decomposition, if the images are above a threshold size. Also, the new adaptive 2D decomposition scheme offers a better location for hiding the watermark. The decomposition will divide the image until accept the sub image that have the overall image information. It is able to detect the statistical properties of local sub image that are used into decomposition image.

Lossless wavelet compression will help to passing the watermark during the compression.

The watermark will be added with the pseudo noise after detect the position to hide it ( by using the Adaptive 2D decomposition scheme) and during the compression processing to avoid the losing by thresholding techniques in wavelet compression processing.

### 14. References

[1] Steven Gortler, Peter Schröder, Michael Cohen, Pat Hanrahan, "Wavelet Radiosity", Computer Graphics, Annual Conference Series (Siggraph'93 Proceedings), pp. 221-230, 1993  
[2] Deborah Berman, Jason Bartell, David Salesin, "Multiresolution Painting and Compositing", Computer Graphics, Annual Conference Series (Siggraph'94

Proceedings), pp. 85-90, 1994

[3] Adam Finkelstein, David Salesin, "Multiresolution Curves", Computer Graphics, Annual Conference Series (Siggraph'94 Proceedings), pp. 261-268, 1994

[4] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsberry, Werner Stuetzle, "Multiresolution Analysis of Arbitrary Meshes", Computer Graphics, Annual Conference Series (Siggraph'95 Proceedings), pp. 173-182, 1995

[5] L. Lippert, M. Gross, "Fast Wavelet Based Volume Rendering by Accumulation of Transparent Texture Maps", Eurographics'95 Conference Proceedings, Computer Graphics Forum Vol. 14, Nr. 3, pp. 431-443, 1995

[6] Charles Jacobs, Adam Finkelstein, David Salesin, "Fast Multiresolution Image Querying", Computer Graphics, Annual Conference Series (Siggraph'95 Proceedings), pp. 277-286, 1995

[7] Zicheng Liu, Steven Gortler, Michael Cohen, "Hierarchical Spacetime Control", Computer Graphics, Annual Conference Series (Siggraph'94 Proceedings), pp. 35-42, 1994

[8] Peter Schröder, Wim Sweldens, "Spherical Wavelets: Efficiently Representing Functions on a Sphere", Computer Graphics, Annual Conference Series (Siggraph'95 Proceedings), pp. 161-172, 1995

[9] Ronald DeVore, Björn Jawerth, Bradley Lucier, "Image Compression Through Wavelet Transform Coding", IEEE Transactions on Information Theory, Vol. 38, Nr. 2, pp 719-746, March 1992

[10] J. M. Shapiro, "Embedded Image Coding using Zerotrees of Wavelet Coefficients", IEEE Transactions on

Signal Processing, Vol. 41, Nr. 12, pp 3445-3462, 1993

[11] Philippe Bakaert, Geert Uytterhoeven, Yves Willems, "An Experiment with Wavelet Image Coding", Proceedings of 11th Spring Conference on Computer Graphics, Bratislava, pp. DM1-6, May 1995

[12] Ahmad Zandi, James Allen, Edward Schwartz, Martin Boliek, "CREW: Compression with Reversible Embedded Wavelets", Preprint from RICOH California Research Center, available from <http://www.crc.ricoh.com/misc/crc-publications.html>

[13] Alain Fournier, Michael Cohen, Tony DeRose, Michael Lounsbery, Leena-Maija Reissell, Wim Sweldens, "Wavelets and their Applications to Computer Graphics", Siggraph'95 Course Notes 26, Los Angeles, August 1995

[14] Charles K. Chui, "An Introduction to Wavelets", Academic Press Inc., Boston, MA 1992

[15] Ingrid Daubechies, "Ten Lectures on Wavelets", Vol. 61 of CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, PA 1992

[16] Jelena Kovacevic, Martin Vetterli, "Nonseparable Multidimensional Perfect Reconstruction Filter Banks and Wavelet Bases for  $R^n$

[17] Ian H. Witten, Radford M. Neal, John G. Cleary, "Arithmetic Coding for Data Compression", CACM Vol. 30, No. 6, pp. 520-540, June 1987

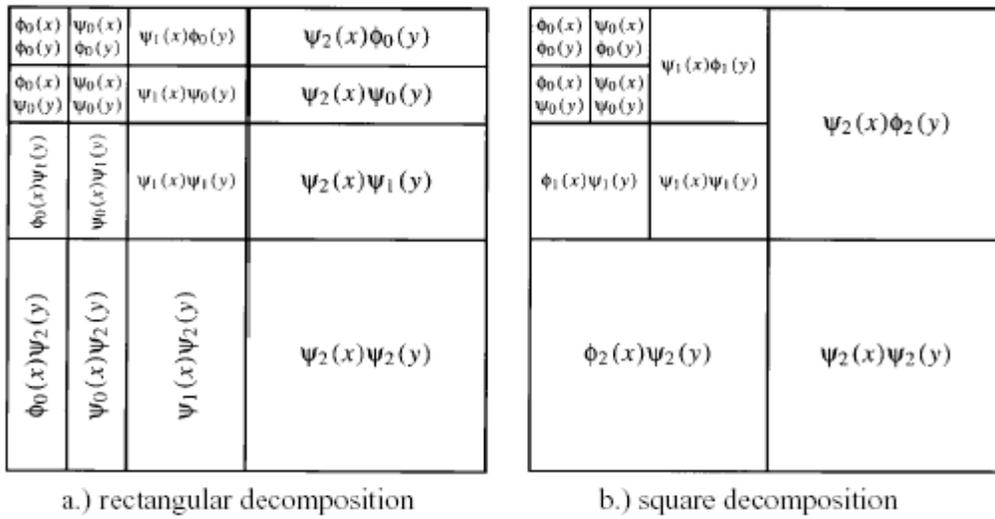


Figure 2 The decomposition types

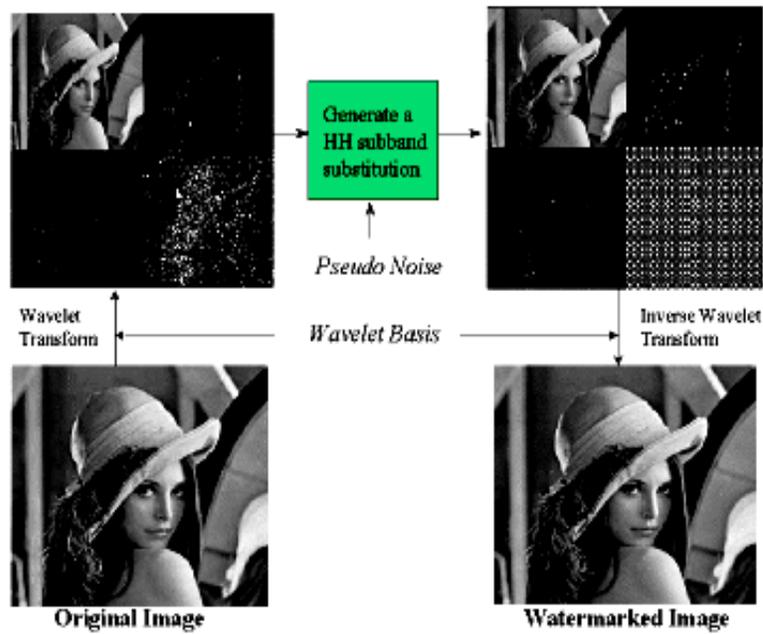


Fig. 3: The watermark embedding process. (The intensity values of the HH subband are magnified by 10 for representation.)

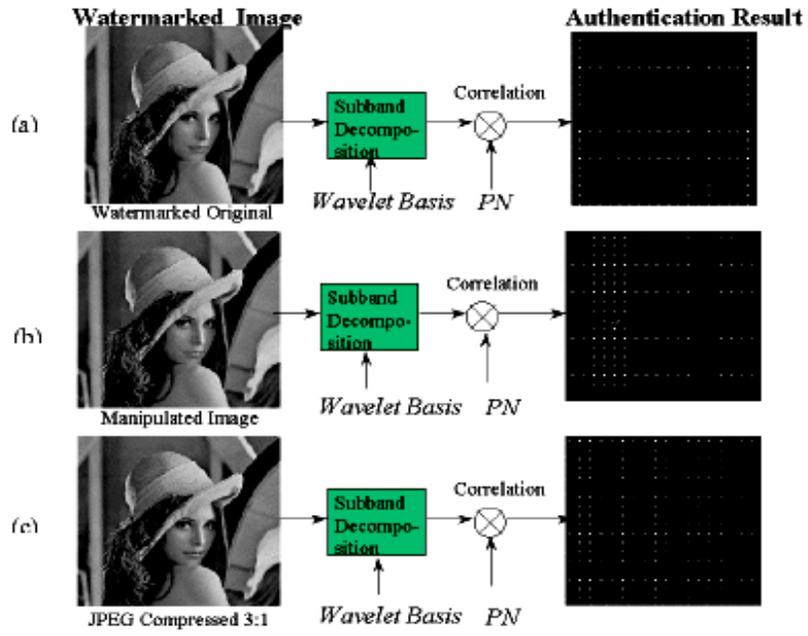


Figure 4: The authentication process and experimental results.