M-band Wavelet Based Pseudo Quantum Watermarking

TONG LIU[1], XUAN XU[2], XIAODI WANG[3]
[1]Department of Math, Tsinghua University, Beijing, CHINA
[2]Department of Chemistry, Hong Kong University of Science and Technology, Hong Kong, CHINA
[3]Department of Math, Western Connecticut State University, Danbury, CT, USA

[1]1808772401@qq.com [2]laoyingtwo@163.com [3]xiaodiwang1@yahoo.com

Abstract: Computational methods derived from digital signal processing are playing a significant role in the security and copyrights of audio, video, and visual arts. In light of the quantum computing, the corresponding algorithms are becoming a new research direction in today's high-technology world. The nature of quantum computer guarantees the security of quantum data, so a safe and effective quantum watermarking algorithm is in demand. Quantum watermarking is the technique that embeds the invisible quantum signal into quantum multimedia data for copyright protection. Different from most traditional algorithms, we propose new algorithms which apply a quantum or a pseudo quantum watermarking in M-band wavelet domain. Assured by the Heisenberg uncertainty principle and quantum no-cloning theorem, the security of quantum watermark can reach a very high-level standard. In other words, these watermarking algorithms can defeat nearly all attackers, no matter using classical computer or quantum computer.

Key-Words: Discrete M-band wavelet transforms (DMWT); Quantum and pseudo watermarks; Quantum computing; Quantum and pseudo Images.

1 Introduction

The purpose of a watermark is to secure the authentication of a multimedia data or visual art work. There are always some unlawful people trying to attack or destroy the watermark by all means possible. So it's important to protect the products and art works from being copied or stolen. The most important indicators of watermarked images are the security and robustness. A watermark with bad security and robustness can be attacked or destroyed easily, which makes it meaningless.

Traditional digital watermarking algorithms based on the Principle Component Analysis (PCA) and wavelet transforms, separately or combined, are used widely on classical computers, such as in [5]-[7]. Also, some algorithms are combined in other ways, such as Discrete Cosine Transform, Discrete Wavelet Transform, and Principle Component Analysis[8],[9]. Of course, these algorithms have good security and robustness on classical computers. However, with the development of quantum computer, traditional watermarking algorithms may not be always safe, especially when facing attacking from (future) quantum computers. Luckily enough, we can solve this problem by the technique of quantum watermarking. Different from most traditional algorithms, we propose a new algorithm which applies quantum watermarking in M-band Wavelet domain. Assured by the Heisenberg uncertainty principle and quantum no-cloning theorem, the security of our quantum watermarking algorithm can reach a high-level standard. In other words, this watermarking algorithm can defeat nearly all attackers, no matter using classical computer or quantum computer.

Recently, a few algorithms for quantum watermarking have been proposed in [1]-[4]. They developed quantum watermarking algorithms based on FRQI (Flexible Representation of Quantum Image)[1]. According to the FRQI, a quantum image's representation can be written as the form shown below:

$$I(\theta) = \frac{1}{2^n}\sum_{i=0}^{2^{2n-1}} |c_i\rangle \otimes |i\rangle$$

where $|c_i\rangle = cos\theta_i|0\rangle + sin\theta_i|1\rangle$ and $|0\rangle, |1\rangle$ are 2-D computational basis, $(\theta_0, \theta_1, \ldots, \theta_{2^{2n-1}})$ is the vector of angles encoding colours $\theta_i \in \left[0, \frac{\pi}{2}\right]$and $|i\rangle$, for i= 0, 1…,2^{2n-1}, are (2^{2n-1})-D computational basis. But there is no explanation that how they used these angles to encode colours at different pixels. To avoid using FRQI we define (*pseudo) quantum image* and create an algorithm based on this definition, then simulate our theory using ordinary computer. Although it still generates pseudo random numbers and the calculation speed is supposed to be much slower than on a quantum computer, the results seem fantastic. Also, our algorithm can get rid of the trouble that may occur when applying FRQI on classical computers: the calculation is too complicated. As far as we know, our attacking experiments showed that our method is much safer with better robustness than that of existing algorithms.

2 Primaries

2.1 Overview of Development of Quantum Computers and Quantum Computing

Quantum computing is often considered one of the most logical successors to traditional computing. If pulled off, it could spur innovation across many fields, from sorting through tremendous Big Data stores of unstructured information — which will be key in making discoveries — to designing super materials, new encryption methods, and drug compounds without trial-and-error lab testing.

For all of this to happen, though, someone has to build a working quantum computer. In past few decades, quantum computers and the corresponding algorithms have been developed rapidly, such as in 2001 IBM built the world's first 7-qubit "quantum computer" to give a demonstration. In 2007, D-wave corporation in Canada announced that they had finished the 16-qubit commercial "quantum computer" for the first time and it was improved to 48-qubit in 2008, and in 2011, D-wave declared their achievement of 128-qubit commercial "quantum computer". In 2013, the Google purchased a 512-qubit D-Wave II System, and has designed a plurality of work on machine learning algorithms which may provide the most creative problem-solving process under the known laws of physics. But scientists argued that these computers are not truly quantum computer yet. The problem is, a quantum computer won't work until you remove what's called quantum decoherence, or errors in calculations due to heat, defects, or electromagnetic radiation. Qubits are extremely delicate; simply measuring one could change its state. You could have a bit-flip error, which simply means the opposite state (1 instead of 0, for example). And you could have a phase-flip error, which is an error in the sign of the superposition state. So in any quantum computing, quantum error correction is a necessary part of any large-scale reliable quantum computer design. But with previous concepts, you could only detect one or the other at the same time. We're a big step closer now, though. In May of 2015, IBM researchers, for the first time, have figured out how to detect and measure both bit-flip and phase-flip quantum errors simultaneously. They also outlined a new, square quantum bit circuit design that could scale to much larger dimensions.

Furthermore, researchers at the University of Sydney and Dartmouth College said they have found a new way to design quantum memory, a key element in realizing quantum computing.

2.2 Quantum Computer and Qubits

A quantum computer is a device for computations that makes direct use of quantum mechanical properties. While normal computation and information are based on classical bits, quantum computation and quantum information are based on quantum bits (qubits). Just like a classical bit, either 0 or 1, a qubit also has a state. Their difference is that a qubit can be in both $|0\rangle=\begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle=\begin{bmatrix}0\\1\end{bmatrix}$ states simultaneously and any linear combinations of them. $|0\rangle$ and $|1\rangle$ are called computational basis states or basis and $|\psi\rangle= a|0\rangle+ b|1\rangle$ is called super position of $|0\rangle$ and $|1\rangle$, where a and b are complex numbers satisfying $|a|^2 + |b|^2 = 1$. Moreover, $|0\rangle$ and $|1\rangle$ form an orthonormal basis for 2-D Hilbert space, a special vector space. We can think the qubit as the following geometric representations (Fig.1), which can be rewritten as the form of qubit: $|\psi\rangle= cos\frac{\theta}{2}|0\rangle+ e^{i\varphi}sin\frac{\theta}{2}|1\rangle$, where θ and φ are real numbers and a qubit defines a point on the unit 3-D sphere.

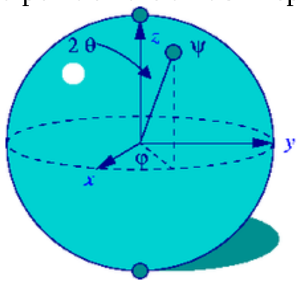

Fig. 1 Qubits

2.3 Discrete M-band Wavelet Transform (DMWT)

Discrete M-Band Wavelet Transform uses a set of M filter banks (M≥2) to break a k-D signal into M^k different frequency levels. Daubechies wavelets are classical 2-Band wavelets. A 4-Band 2-Dwavelet transform decomposes an image into one approximation (low frequency) component and 15 detail (high frequency) components. The 2-D discrete M-Band wavelet transform of an image matrix I is done by multiplying a wavelet transform matrix to the left side of input image, and then by the transpose to the right side, written as TIT^t, where T is the wavelet transform matrix which is

ISBN: 978-1-61804-328-3

orthonormal and T^t is the transpose of T and hence $T^t = T^{-1}$.

In order to apply DMWT to a colour image, we decompose the RGB-mode colour image into three matrices, I_1, I_2, and I_3 for red, green, and blue, respectively. We then apply DMWT to each one of them to obtain

$$I'_k = TI_kT^t$$

for $k = 1,2,3$, respectively. So the transformed image can be formed by combination of I'_1, I'_2, and I'_3. An example of 4-band wavelet transform matrix T is given below:

$$T = \begin{bmatrix}
\alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 & \alpha_7 & \alpha_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 & \alpha_7 & \alpha_8 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 & \alpha_7 & \alpha_8 \\
\alpha_5 & \alpha_6 & \alpha_7 & \alpha_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\
\beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & \beta_7 & \beta_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & \beta_7 & \beta_8 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & \beta_7 & \beta_8 \\
\beta_5 & \beta_6 & \beta_7 & \beta_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 \\
\gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6 & \gamma_7 & \gamma_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6 & \gamma_7 & \gamma_8 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6 & \gamma_7 & \gamma_8 \\
\gamma_5 & \gamma_6 & \gamma_7 & \gamma_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\
\delta_1 & \delta_2 & \delta_3 & \delta_4 & \delta_5 & \delta_6 & \delta_7 & \delta_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \delta_1 & \delta_2 & \delta_3 & \delta_4 & \delta_5 & \delta_6 & \delta_7 & \delta_8 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \delta_1 & \delta_2 & \delta_3 & \delta_4 & \delta_5 & \delta_6 & \delta_7 & \delta_8 \\
\delta_5 & \delta_6 & \delta_7 & \delta_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \delta_1 & \delta_2 & \delta_3 & \delta_4
\end{bmatrix}$$

where

α= [-0.06737176, 0.09419511, 0.40580489, 0.56737176, 0.56737176, 0.40580489, 0.09419511, -0.06737176].

β= [-0.09419511, 0.06737176, 0.56737176, 0.40580489, -0.40580489, -0.56737176, -0.06737176, 0.09419511].

γ= [-0.09419511, -0.06737176, 0.56737176, -0.40580489, -0.40580489, 0.56737176, -0.06737176, -0.09419511].

δ= [-0.06737176, -0.09419511, 0.40580489, -0.56737176, 0.56737176, -0.40580489, 0.09419511, 0.06737176].

It's easy to verify that

$$\sum_{i=1}^{8} \alpha_i = \sqrt{4} = 2, \sum_{i=1}^{8} \beta_i = \sum_{i=1}^{8} \gamma_i = \sum_{i=1}^{8} \delta_i = 0,$$

$$|\boldsymbol{\alpha}| = |\boldsymbol{\beta}| = |\boldsymbol{\gamma}| = |\boldsymbol{\delta}| = 1,$$

$$\boldsymbol{\alpha} \cdot \boldsymbol{\beta} = \boldsymbol{\alpha} \cdot \boldsymbol{\gamma} = \boldsymbol{\alpha} \cdot \boldsymbol{\delta} = \boldsymbol{\beta} \cdot \boldsymbol{\gamma} = \boldsymbol{\beta} \cdot \boldsymbol{\delta} = \boldsymbol{\gamma} \cdot \boldsymbol{\delta} = 0$$

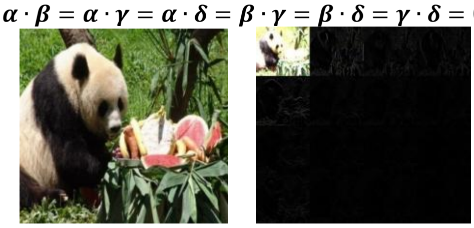

Fig. 2 Original signal and its corresponding wavelet transformed signal. In the right picture, left top part is approximation, others are details.

3 Watermarking Procedure

3.1 Watermark Embedding Procedure

Let I be the original picture of $4^n \times 4^n$, and J be the watermark of $4^{n-1} \times 4^{n-1}$. The following procedures are applied to all the 3 colour matrices mentioned in the section 2.3.

Step 1.1 (DMWT): Apply DMWT to the original image I to get $I_T = TIT^T$, the wavelet transform of I in the M-band wavelet domain. let I' be the approximation part of I_T.

Step 1.2 (Signal conversion): Let $u = \max(I'_{mn})$, and $v = \min(I'_{mn})$. We apply a linear transformation F to I' and obtain

$$\theta_{mn} = \frac{\pi(I'_{mn} + \mathrm{u} - 2\mathrm{v})}{6(\mathrm{u} - \mathrm{v})}$$

so that θ_{mn} is in the interval $[\pi/6, \pi/3]$.

Step 1.3 Apply a linear transformation F_w to a watermark J and obtain

$$\alpha_{mn} = \frac{\pi}{1530} J_{mn} + \frac{\pi}{6}$$

so that α_{mn} is also in the interval $[\pi/6, \pi/3]$.

Step 1.4 (Watermark Embedding): For each pixel (m, n), define random qubit $|k_{mn}\rangle = P_1|0\rangle + P_2|1\rangle$, if $|P_1|>|P_2|$, then

$$\theta_{w,mn} = \cos^{-1}(\cos\theta_{mn} + \varepsilon\cos\alpha_{mn})$$

Else

$$\theta_{w,mn} = \sin^{-1}(\sin\theta_{mn} + \varepsilon\sin\alpha_{mn})$$

Where ε is the embedding intensity indicator. Save all the random qubits into a codebook K.

Step 1.5 (Transform signal to original domain): Apply the inverse of F:

$$I'_{w,mn} = \frac{6(\mathrm{u} - \mathrm{v})\theta_{w,mn}}{\pi} + 2v - u$$

Step 1.6 (Inverse DMWT): Replace the approximation part of I_T of I', by I'_w to obtain I_{wT}. Then apply the inverse wavelet transform to I_{wT} to get $I_w = T^T I_{wT} T$. Finally, I_w represents the watermarked image.

3.2 Watermarks extracting procedure

Steps 2.1&2.2 are the same as 1.1&1.2, for we have to change the normal image into quantum signals. Then we apply Steps 1.1&1.2 to watermarked image I_w to get θ_w.

Step 2.3(Watermark extracting) Call the "code book" K, for each pixel, we have $|k_{mn}\rangle = P_1|0\rangle + P_2|1\rangle$, if $|P_1| > |P_2|$, then

$$\alpha_{e,mn} = \cos^{-1}\left(\frac{\cos\theta_{w,mn} - \cos\theta_{mn}}{\varepsilon}\right)$$

else

$$\alpha_{e,mn} = \sin^{-1}\left(\frac{\sin\theta_{w,mn} - \sin\theta_{mn}}{\varepsilon}\right)$$

Step 2.4 (Transform signal to original domain) Apply inverse of F_w:

$$J_{e,mn} = \frac{1530}{\pi}\alpha_{e,mn} - 255$$

then J_e is the extracted watermark.

3.3 Notes to the Watermarking Procedure

ISBN: 978-1-61804-328-3

3.3.1 Pseudo quantum signals

In Steps 1.2&1.3, we defined linear transformations F& F_w. We call them "pseudo quantum signal converters", as we can change the classical signals into the *form* of quantum signals through them. We then define qubit $|i_{mn}\rangle = \cos\theta_{mn}|0\rangle + \sin\theta_{mn}|1\rangle$, where θ_{mn} is related to the pixel values after the linear transformations For F_w. This transform changes the pixel values into the form of angles, thus we can define corresponding qubits to represent the signal.

Definition 3.1 The corresponding qubits above $|i_{mn}\rangle = \cos\theta_{mn}|0\rangle + \sin\theta_{mn}|1\rangle$are called *"pseudo quantum signals".*

After all, they're not exactly the same as real quantum signals, so we call them "pseudo quantum signals". This can help us simulate quantum signals and quantum computing in situations where quantum computers are not available. If we carry out the algorithm on a quantum computer, we can directly change the classical signals into quantum signals without Steps 1.2&1.3.

3.3.2 Estimation of the embedding intensity ε

In Step 1.4, we embed a watermark according to the embedding intensity ε. But only some of the ε values can be used. Note that the definition domain of the function $f(x) = \cos^{-1} x$ is [0, 1], according to the algorithm, we must have

$$\cos\theta_{w,mn} = \cos\theta_{mn} + \varepsilon\cos\alpha_{mn} \le 1$$

therefore

$$\varepsilon \le \frac{1-\cos\theta_{mn}}{\cos\alpha_{mn}}$$

If $\theta_{mn} \to 0$, then we have $\sup\varepsilon \to 0$. But if $\varepsilon$is too small, then the watermark extracting will be very difficult (note that in the Step 2.3 we will divide a number by ε, the smaller $\varepsilon$is, the greater the error will be). Considering this problem and that in the situation of sine, in Steps 1.2&1.3, we control the definition domain of θ_{mn} and $\alpha_{mn}$to $[\pi/6, \pi/3]$ .Therefore both $\cos\theta_{mn}$and$\cos\alpha_{mn} \in [1/2, \sqrt{3}/2]$, and thus

$$\varepsilon \le \frac{1-\cos\theta_{mn}}{\cos\alpha_{mn}} \le \frac{1-\sqrt{3}/2}{\sqrt{3}/2} = \frac{2\sqrt{3}}{3} - 1 < 0.155$$

3.3.3 Random qubits

In Step 1.4, whether we use the way of cosine or sine depends on the random qubits. Article [8] proposed a quantum watermarking algorithm, but they only use random numbers 0 or 1. Of course this way is widely used on classical computers. But on quantum computers, we can't ensure that all the random qubits are $|0\rangle$ or $|1\rangle$. However, we can use the probability that $|0\rangle$ or $|1\rangle$ occurs. Note that we have random qubits $|k_{mn}\rangle = P_1|0\rangle + P_2|1\rangle$, if $|P_1| > |P_2|$, this is equivalent to$|P_1|^2 > |P_2|^2$, hence the probability that $|0\rangle$ occurs is greater than $|1\rangle$ occurs. Without measurement (or the random qubits will be broken, and we can't legally extract the watermark), we directly regard such a qubit as $|0\rangle$, else $|1\rangle$. If someone wants to steal the codebook, he or she has to measure the qubits, due to the Heisenberg uncertainty principle and quantum no-cloning theorem, getting all the data $|0\rangle$ or $|1\rangle$ correctly is impossible.

In our experiments with a classical computer, we used a set of pseudorandom numbers $r_{mn} \in [0, 1]$. If $r_{mn} > 1/2$, we embed the watermark pixel value at (m, n) into cosine portion of the *pseudo quantum image* at the same location, else we insert it into corresponding sine portion as we explained in the Section 3.1 Step 1.4.

4 Conclusion

4.1 Watermark Embedding and Extracting

We use different original images and embed the watermark that is the logo of *Affiliated High School to Jilin University*, the high school Tong and Xuan graduated from, and the results are shown below:

Fig.3 The first two pictures are watermarked images, and the last two are corresponding extracted watermarks.

4.1.1 PSNR

To measure the quality of the watermarked image, we use PSNR (Peak Signal-to-Noise Ratio).The signal in this case is the original image, and the noise is the error between original image and watermarked image. Normally, the higher PSNR means the better quality of watermarked image.

Definition 4.1 Given a noise-free $m \times n$ image (original image) I and its noisy approximation K (watermarked image), *MSE* (Mean Square Error) is defined as:

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(I_{ij} - K_{ij})^2$$

(For colour images with three RGB values per pixel, the MSE is the sum over all squared value differences divided by image size and by 3).

Definition 4.2 PSNR = 10*log (255^2/*MSE*).

Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. For classical watermark algorithms, the PSNR is usually 50~70, and for compression, 30~40.

In our experiment, the PSNRs of the watermarked images range from 66.524 to 120.34 according to embedding intensity ε. They're much higher than existing watermark algorithms as far as we know. So our watermark algorithm is proven to be of higher quality,

4.1.2 Relative Similarity (RS)

Definition 4.3 For imagesI_1, I_2, the relative similarity (RS) of I_2 to $I_1$is defined by:

$$RS(I_2, I_1) = 1 - \frac{||I_2 - I_1||_1}{||I_1||_1}$$

where $||A||_1$ is the 1-norm of matrix A. For colour images, we calculate the RS for red, green and blue matrices, respectively, and RS is the average of them.

The formula above shows that the RS of two same images is 1, and when RS is closer to 1, two images are more similar to each other. In other words, the indistinguishability of watermarked image will be achieved as long as $RS \approx 1$.

The following table shows the PSNR and RS for watermarked "panda" with different embedding intensity.

No.	1	2	3	4	5
ε	0.001	0.002	0.005	0.01	0.02
PSNR	120.34	109.44	91.407	77.682	66.524
RS	0.9940	0.9888	0.9726	0.9445	0.8908

Note that in our experiment, if ε >0.02, then RS<0.9.So the RS will be low for biggerε. If ε is too small, although we have higher PSNR and RS, the robustness will be loosen. Our experiments show that the robustness will go down if ε <0.002. So ε =0.01, 0.005 or between two are suitable values with relatively high PSNR and RS values.

4.2 Attacking and Extracting

In this research we programmed on Matlab® to test our algorithms. The results show that the security of our algorithm is excellent as explained in previous sections. To check the robustness of the watermark, here we need to simulate attacking watermarked images, and then check if we can extract the watermark successfully, and whether it's effective. The followings are attacked images and statistics results. The examples of attacked images are based on Gaussian noise, salt &pepper noise, speckle noise and compression, respectively.

Fig.4 Attacking: (from left) Gaussian 0.01; Salt & Pepper 0.05; Speckle 0.05; Compression 1:4. Their PSNRs are: 28.9157; 40.2471; 30.3108; 33.0310

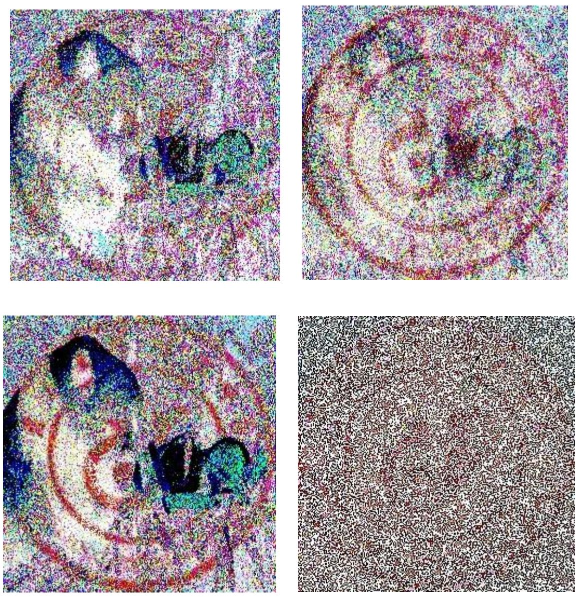

Fig.5 Extracted watermark corresponding to Fig.4. Their PSNRs are: 28.0025; 28.3381; 28.9258; 28.5515

ISBN: 978-1-61804-328-3

In fact, a slight attack can cause a distortion of watermarked image. Although the PSNR of the extracted watermark is not very high, we can still recognize some features of the watermark. Normal process will not affect the watermarked image from extracting the recognizable watermark. And all the heavy attacking ways will destroy watermark, but the same time they will cause a great loss of information resulting in a low quality image. So the experiments show that our algorithm can defeat general attacking.

Note that in our algorithm, pixels are encrypted respectively, so attacking one pixel will only affect itself. Moreover, compared with the algorithm we came out in our previous research, which used more than 10 minutes to embed a 729×729 image, this new algorithm needs no more than 10 seconds on the same personal computer to embed a 1024×1024 image, and with a quantum computer this process must be much faster. So we can now consider about embedding a video with a watermark.

5 Concluding Remarks and Future Research

In this paper, we present a new pseudo quantum watermarking approach in M-band Wavelet Domain. We have shown the efficiency in applying our method for performing watermark embedding and verification. As a result, the watermarked image with such a well-chosen embedding domain is much safer and much difficult to attack. Our computational time is also much shorter than that of our previous research.

This algorithm can be carried out on both classical computers and quantum computers, so it can be widely used for audio, video, and still image file's encryption, security information transmission, etc. We defined *RS* to represent the quality of indistinguishability. In our future research, we will derive a mathematical formula to determine a threshold $\varepsilon>0$ for each pre-determined *RS*. We may carry out a better algorithm that uses less time so that we can divide a video signal into still images and embed a watermark into each image. Also we'll be able to use our algorithm to embed a watermark image into an audio signal.

However, though our algorithm is used for copyright protection, our biggest hope is that there would be no pirates all over the world.

References:

[1] P.Q. Le, F. Dong, K. Hirota, A flexible representation of quantum images for polynomial preparation, image compression and processing operations, *Journal of Quantum Information Processing* (2010), doi:10.1007/s11128-010-0177-y.

[2] A.M. Iliyasu et al, Watermarking and authentication of quantum images based on restricted geometric transformations, *Inform. Sci.* (2011), doi:10.1016/j.ins.2011.09.028.

[3] P.Q. Le, A.M. Iliyasu, F. Dong, K. Hirota, *Efficient color transformations on quantum images, Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII)*, 15 (6) (2011) pp. 698–706.

[4] Wei-Wei Zhang, FeiGao, Bin Liu, Heng-YueJia, Qiao-Yan Wen, Hui Chen, A Novel Watermark Strategy For Quantum Images, http://www.paper.edu.cn

[5] P. Steffen, P.N. Heller, R.A. Gopinath, and C.S Burrus, "Theory of regular m-band wavelet bases," *IEEE Trans. Signal Processing*, Vol. 41, pp. 3497-3511, Dec. 1993.

[6] A. K. Mostafa, A. S. Tolba, F. M. Abdelkader, H.M. Elhind, "Video Watermarking Scheme Based on Principal Component Analysis and Wavelet Transform," *Int. J. of Computer Science and Network Security*, Vol.9, pp. 45-52, Aug. 2009.

[7] N.I. Yassin, N. M. Salem, and M. I. El Adawy, "Block based video watermarking scheme using wavelet transform and principle component analysis," *Int. J. of Computer Science*, Vol. 9, Issue 1, pp. 296-301, Jan. 2012.

[8] A. Singh, and A. Tayal, "Performance analysis of digital image watermarking using discrete wavelet transform, discrete cosine transform and singular value decomposition based on PSNR and NC," *2012 Proc. Int. Conf. on Computing and Control Engineering, ICCCECS 730*, April 2012.

[9] Tong Liu, Xuan Xu, Xiaodi Wang, "M-band Wavelet and Cosine Transform Based Watermark Algorithm Using Randomization and Principle Component Analysis", *International Journal of Science and Engineering Investigations*, vol. 2, issue 13, pp. 1-4, February, 2013.