Quantum Watermarking in M-band Wavelet Domain

Authors: Xuan Xu, Tong Liu

School: Affiliated High School of Jilin University

Changchun City, Jilin Province, China

School teacher: Yulian Ren

Advisor: Xiaodi Wang

All authors made equal contribution to this research project and the names list above are in alpha beta order.

Quantum Watermarking in a M-band Wavelet Domain

Abstract— Computational techniques derived from digital signal processing are playing a significant role in the security and digital 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. This research presents a quantum watermarking algorithm in discrete M-band wavelet domain. 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 quantum watermark can reach a very high-level standard. In other words, this watermarking algorithm can defeat nearly all attackers, no matter using classical computer or quantum computer.

Keywords—Discrete M-band wavelet transforms (DMWT); Quantum watermark; Quantum computing; Quantum Image

Background introduction— Recently, the hardware of quantum computers and quantum computing theory are developed rapidly. In 2013, Google purchased a 512-qubit quantum computing system and planned to upgrade to 2048 qubits in the near future. Also, the algorithms of solving linear system of equations using quantum computer has been accomplished. So the quantum computing era is coming and this also requires us to solve the problems of the security of quantum data.

Traditional digital watermarking algorithms, based on the principle component analysis (PCA) and discrete wavelet transforms (DWT), separately or combined, are used widely on classical computers. Also, some algorithms are combined with DWT, PCA and DCT. Of course, these algorithms have good security and robustness on classical computers. However, with the development of quantum computer, the 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.

Highlights of our research— We applied quantum watermarking in M-band wavelet domain, which combines the advantages of both classical algorithms and quantum algorithms. Thus it should have a higher quality of security and robustness. We defined *pseudo quantum signals* and corresponding transformations, so we can simulate the quantum computing by implementing our algorithm on a classical computer. We also defined a new way to represent the quality of indistinguishability of a watermarked image—*Relative Similarity (RS)* and examined it by our experiments. Our attacking experiments showed that our method is much safer with better robustness than that of existing algorithms as far as we know. Furthermore, we wrote all computer source code ourselves. Compare this research with our previous work, the computational speed of our new algorithm has also increased dramatically: from over ten minutes to less than ten seconds, though we just used a classical computer to simulate our quantum algorithm.

Quantum Watermarking in M-band Wavelet Domain

1. Introductions

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. But according to their purpose, after their attacking, the quality of the watermarked multimedia data or visual art works can't be too low. This limits their attacking intensity. 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 [13]-[15]. Also, some algorithms are combined in other ways, such as Discrete Cosine Transform [15], [16] or Principle Component Analysis [13], [14], [16]. 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 [7],[8],[17]. They developed quantum watermarking algorithms based on FRQI (Flexible Representation of Quantum Image) [2].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 $|c_i\rangle = \ cos\theta_i|0\rangle + \ sin\theta_i|1\rangle$, $|0\rangle$ and $|1\rangle$ are 2-D computational basis, (θ_0, θ_1, ... ,$\theta_{2^{2n-1}}$) is the vector of angles encoding colors $\theta_i \in [0, \pi/2\ ]$, and $|i\rangle$, for *i*= 0, 1…, 2^{2n-1}, are (2^{2n-1})-D computational basis. But there is no explanation on how they used these angles to encode colors 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

In past few decades, the quantum computers and the corresponding algorithms have been developed rapidly. 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. In 2011, D-wave declared their achievement of 128-qubit commercial quantum computer. The company had set goals for developing 1024-qubit quantum computer and proposed quantum computer’s further applications. D-wave’s development implies that the principle of quantum computer has been matured and the practical technology also has made substantial progress. 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. They also plan to upgrade the quantum computer to 2048-qubit in few years. In the meanwhile, the quantum optics and quantum information research team led by Professor Pan Jianwei of University of Science & Technology of China, for the first time in the world achieved a successful implementation of the quantum computer for solving a linear system of equations in an experiment. According to their theory, using the quantum computer in GHz clock frequency it will take only 10 seconds to solve the linear system of equations with 10^{24} unknowns. If we use classical computers, it will need hundreds of years to finish the same job. 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 Some preliminaries about quantum computing and quantum image

2.2.1 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 a 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. Since there are infinitely many points on the unit sphere, one could store an entire text of *the Three Kingdoms* in the infinite binary expansion of θ[1].

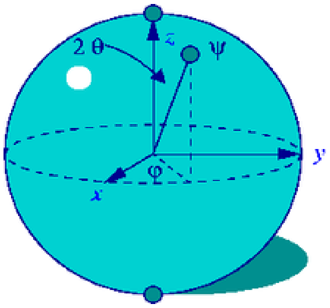


Fig. 1 Qubits

2.2.2 Measurements

Quantum measurements are described by a collection of measurement operators $\{M_k\}$. These are operators acting on the state space of the system being measured. The index k refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that result k occurs is given by: $p(k) = \langle\psi|M_k^T M_k|\psi\rangle$, and $\sum_{k=0}^{n} M_k^T\, M_k = I$. Simply, for $n = 1, M_0 = \ |0\rangle\langle 0| = \begin{bmatrix}1\\0\end{bmatrix}[1 \quad 0] = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}$ and $M_1 = |1\rangle\langle 1| = \begin{bmatrix}0\\1\end{bmatrix}[0 \quad 1] = \begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$. Therefore, we have $M_0^T M_0 + M_1^T M_1 = I$. Let $|\psi\rangle=a|0\rangle+b|1\rangle$. Note that $M_0^T M_0 = M_0$, hence $p(0) = \langle\psi|M_0^T M_0|\psi\rangle = \langle\psi|M_0|\psi\rangle = [\bar{a} \quad \bar{b}]\begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\begin{bmatrix}a\\b\end{bmatrix} = |a|^2$. Similarly $p(1) = |b|^2$.[1]

2.3 Discrete M-band Wavelet Transform (DMWT)

Discrete M-Band Wavelet Transform uses a set of M filter banks ($M \geq 2$) to break a k-D signal into M^k different frequency levels. Daubechies wavelets are classical 2-Band wavelets. A 4-Band 2-D wavelet 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 a 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 orthonormal and T^t is the transpose of T and hence $T^t = T^{-1}$.

In order to apply DMWT to a color image, we decompose the RGB-mode color 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

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

$$|\boldsymbol{\alpha}| = |\boldsymbol{\beta}| = |\boldsymbol{\gamma}| = |\boldsymbol{\delta}| = 1, \boldsymbol{\alpha} \cdot \boldsymbol{\beta} = \boldsymbol{\alpha} \cdot \boldsymbol{\gamma} = \boldsymbol{\alpha} \cdot \boldsymbol{\delta} = \boldsymbol{\beta} \cdot \boldsymbol{\gamma} = \boldsymbol{\beta} \cdot \boldsymbol{\delta} = \boldsymbol{\gamma} \cdot \boldsymbol{\delta} = 0$$[12]

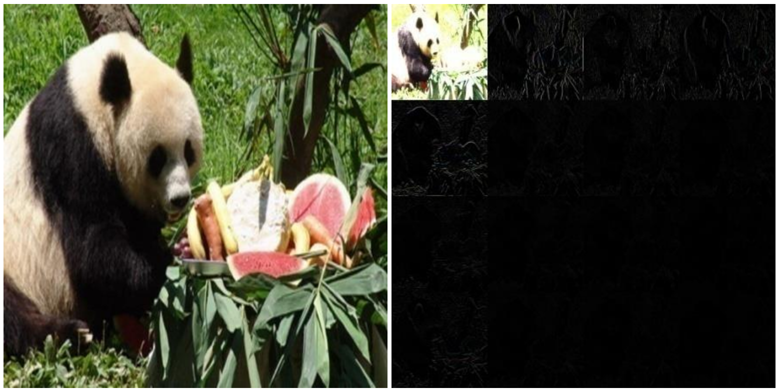


Fig. 2 Original signal and its corresponding wavelet transformed signal. In the right picture, left top part is approximation, others are details.

2.4 Matrix norm

A vector's p-norm is defined as:

$$||X||_p = \left(\sum_{i=1}^{n} |x_i^p|\right)^{1/p}$$

The matrix's operator norm corresponding to the *p*-norm for vectors is given by:

$$||A||_p = \max_{x\neq 0} \frac{||Ax||_p}{||x||_p}$$

These are different from the entry wise *p*-norms and the Schatten *p*-norms for matrices treated below, which are also usually denoted by $||A||_p$.

In the case of *p*=1, the norm can be computed as

$$||A||_1 = \max_{1\leq i\leq n} \sum_{j=1}^{m} |a_{ij}|$$

and it's simply the maximum absolute row sums of the matrix *A*.

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 color matrices mentioned in the section 2.3.

Step 1.1 (DMWT): Apply DMWT to the original image I to get $I_T = TIT^T$, 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: $I'_{w,mn} = \frac{6(\mathrm{u-v})\theta_{w,mn}}{\pi} + 2v - u$

Step 1.6 (Inverse DMWT): Replace the approximation part of I_T, 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 Watermark 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 $\alpha_{e,mn} = \cos^{-1}(\frac{\cos\theta_{w,mn} - \cos\theta_{mn}}{\varepsilon})$, else $\alpha_{e,mn} = \sin^{-1}(\frac{\sin\theta_{w,mn} - \sin\theta_{mn}}{\varepsilon})$.

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

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 F or F_w. This transform changes the pixel values into the form of angles, thus we can define corresponding qubits to represent the signal.

Definition 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 valid. Note that the definition domain of the function $f(x) = \cos^{-1} x$ is [0, 1], according to our algorithm, we must have $\cos\theta_{w,mn} = \cos\theta_{mn} + \varepsilon\cos\alpha_{mn} \leq 1$; therefore $\varepsilon \leq \frac{1-\cos\theta_{mn}}{\cos\alpha_{mn}}$. If $\theta_{mn} \to 0$, then we have $\sup\varepsilon \to 0$. But if ε 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 ε 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 α_{mn} to $[\pi/6, \pi/3]$.

Now we can estimate ε: Note that $\cos\theta_{mn} \in [1/2, \sqrt{3}/2]$, $\cos\alpha_{mn} \in [1/2, \sqrt{3}/2]$, thus $\varepsilon \leq \frac{1-\cos\theta_{mn}}{\cos\alpha_{mn}} \leq \frac{1-\sqrt{3}/2}{\sqrt{3}/2} = \frac{2\sqrt{3}}{3} - 1 < 0.155$.

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 that of $|1\rangle$ 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 pseudo random 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. Experiment result

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*, and the results are shown below:

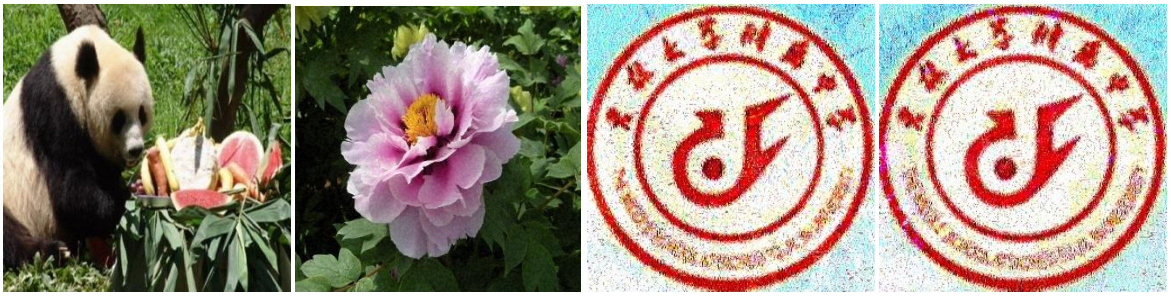


Fig.3The first two pictures are watermarked images, and the last two are corresponding extracted watermarks.

4.1.1 PSNR

To measure the quality of the watermarked image, we use PSNR (Peak Signal-to-Noise Ratio) [10].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 2 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 color images with three RGB values per pixel, the MSE is the sum over all squared value differences divided by image size and by 3)[11].

Definition 3 PSNR = 10*log (255^2/*MSE*). [11]

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 66.5245 to 120.3474 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)

Definition4 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 color 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.

Experiment No.	1	2	3	4	5
ε	0.001	0.002	0.005	0.01	0.02
PSNR	120.3474	109.4471	91.4072	77.6827	66.5245
RS	0.9940	0.9888	0.9726	0.9445	0.8908

Note that in our experiment, if $\varepsilon > 0.02$, then RS < 0.9. So the RS will be low for bigger ε. 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 $\varepsilon < 0.002$. So if ε is between 0.01 and 0.005 then we have 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.

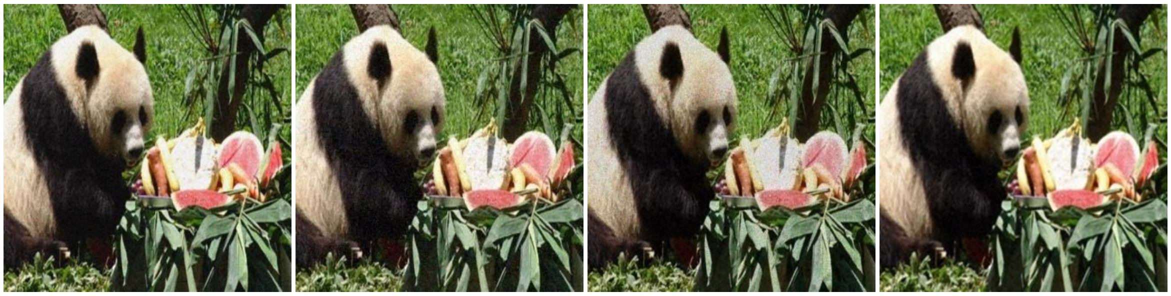


Fig.4 Attacking:(from left) Gaussian 0.01; Salt & Pepper 0.05; Speckle 0.05; Compression 1:4. Their PSNRs are: 28.9157; 40.2471; 30.3108; 33.0310

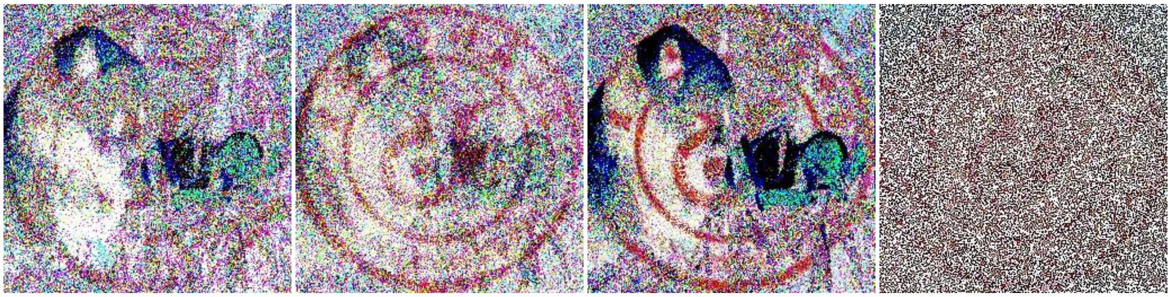


Fig.5 Extracted watermark corresponding to Fig.4. Their PSNRs are: 28.0025; 28.3381; 28.9258; 28.5515

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 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.

6. Acknowledgements

We would like to thank Dr. Xiaodi Wang from Western Connecticut State University, Dr. Zhenji Cui, the principal of Affiliated High School to Jilin University, Yulian Ren, the student research director at Affiliated High School to Jilin University, and Jilin University for their incredible academic and financial supports. Without their constant encouragement and necessary funding, to complete this project would be impossible.

References

[1] M. Nielsen, I. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, New York, 2000.

[2] 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.

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

[4]P.Q. Le, A.M. Iliyasu, F. Dong, K. Hirota, Fast geometric transformations on quantum images, IAENG International Journal of Applied Mathematics 40 (3)(2010) 113–123.

[5]P.Q. Le, A.M. Iliyasu, F. Dong, K. Hirota, Strategies for designing geometric transformations on quantum images, Theoretical Computer Science 412 (2011) 1406–1418.

[6]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) 698–706.

[7]S.E. Venegas-Andraca and S. Bose, Storing, Processing and Retrieving an Image using Quantum Mechanics, Proceedings of the SPIE Conference on Quantum Information and Computation pp. 137-147 (2003).

[8]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

[9]Bourbaki, Nicolas (1987). "Chapters 1–5". Topological vector spaces. Springer. ISBN 3-540-13627-4.

[10]Huynh-Thu, Q.Ghanbari, M. (2008)."Scope of validity of PSNR in image/video quality assessment". Electronics Letters 44 (13): 800–801. doi:10.1049/el:20080522. Edit

[11]"Image Processing Science calculating RMSE and PSNR for color images". Retrieved 6 April 2011.

[12]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.

[13]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.

[14]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.

[15]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,” in 2012 Proc. Int. Conf. on Computing and Control Engineering, ICCCECS730, April 2012.

[16]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.

[17] 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.

Appendix A

Source Code and Experiments

1) M-band Wavelet Transform

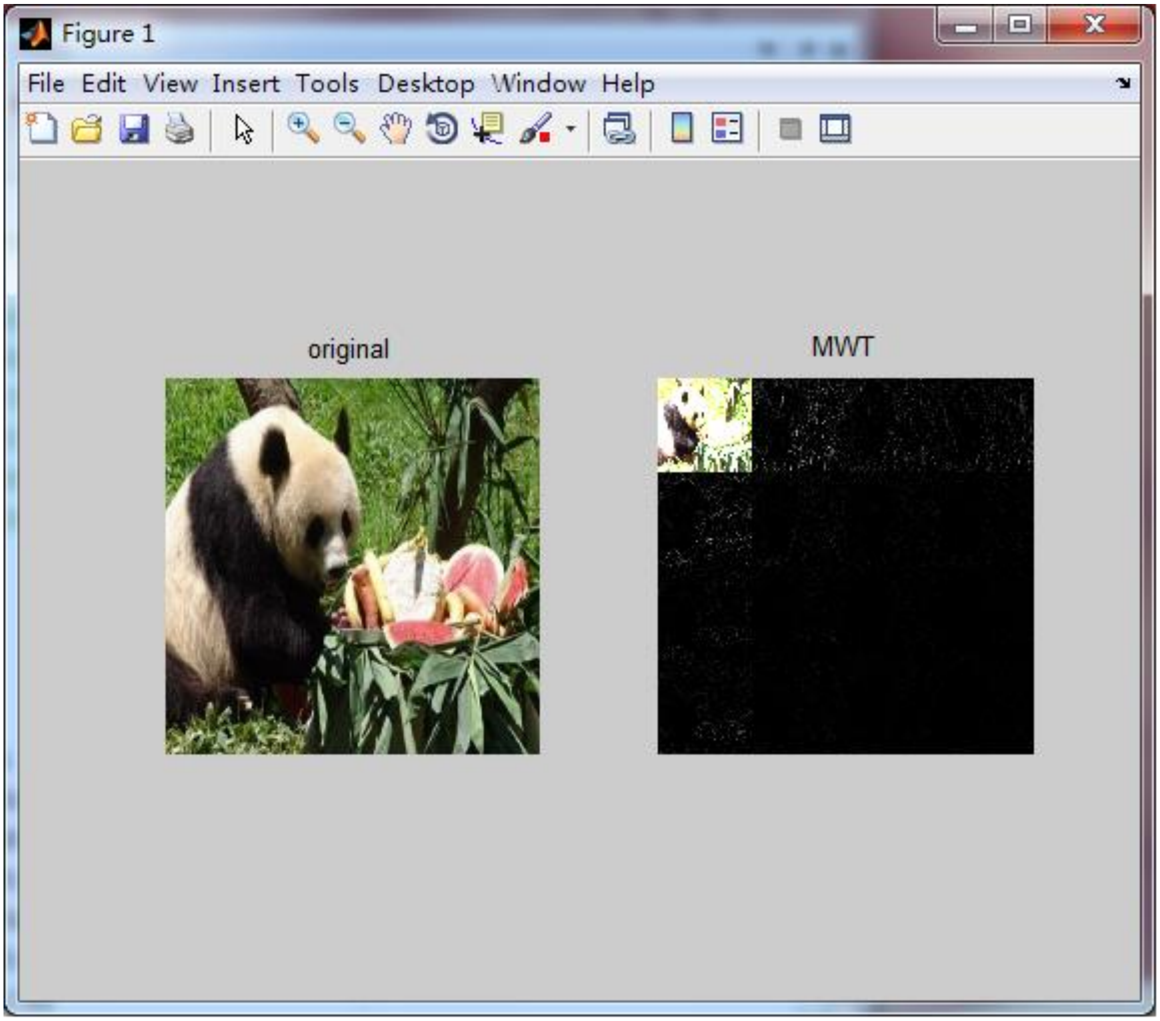
```
clc;clear
format long
v=[-0.067371764 0.094195111 0.40580489 0.567371764
    0.567371764 0.40580489 0.094195111 -0.067371764];
w1=[-0.094195111 0.067371764 0.567371764 0.40580489
    -0.40580489 -0.567371764 -0.067371764 0.094195111];
w2=[-0.094195111 -0.067371764 0.567371764 -0.40580489
    -0.40580489 0.567371764 -0.067371764 -0.094195111];
w3=[-0.067371764 -0.094195111 0.40580489 -0.567371764
```

```
    0.567371764 -0.40580489 0.094195111 0.067371764];

% four vecters which are written as alpha, beta, gamma and delta in essay

x=4;n=5;

% x stands for the band and n stands for the times as 4^5=1024
% A stands for the low frequency and B, C, and D for high

rows = x^(n-1);
cols = x^(n+0);

row = zeros(rows*2*x,1);
col = zeros(rows*2*x,1);
val = zeros(rows*2*x,1);
pos = +0;

for j = 1 : x^(n-1)
    if (j < x^(n-1))

    for k = max(1,x*(j-1)+1) : min(x^n,x*(j-1)+2*x)
        pos = pos+1;
        row(pos) = j;
        col(pos) = k;
        val(pos) = v(k-x*(j-1));
    end

    else

    for k=1:x^n
        if k<=x
            pos = pos+1;
            row(pos) = j;
            col(pos) = k;
            val(pos) = v(k+x);
        elseif k>= max(1,x*(j-1)+1)
```

```
            pos = pos+1;
            row(pos) = j;
            col(pos) = k;
            val(pos) = v(k-x*(j-1));
        end
    end
    end
end

row = row(1:pos);
col = col(1:pos);
val = val(1:pos);

A = sparse(row,col,val,rows,cols);

rows = x^(n-1);
cols = x^(n+0);

row = zeros(rows*2*x,1);
col = zeros(rows*2*x,1);
val = zeros(rows*2*x,1);
pos = +0;

for j = 1 : x^(n-1)
    if (j < x^(n-1))

    for k = max(1,x*(j-1)+1) : min(x^n,x*(j-1)+2*x)
        pos = pos+1;
        row(pos) = j;
        col(pos) = k;
        val(pos) = w1(k-x*(j-1));
    end

    else

    for k=1:x^n
```

```
        if k<=x
            pos = pos+1;
            row(pos) = j;
            col(pos) = k;
            val(pos) = w1(k+x);
        elseif k>= max(1,x*(j-1)+1)
            pos = pos+1;
            row(pos) = j;
            col(pos) = k;
            val(pos) = w1(k-x*(j-1));
        end
    end
    end
end

row = row(1:pos);
col = col(1:pos);
val = val(1:pos);

B = sparse(row,col,val,rows,cols);
rows = x^(n-1);
cols = x^(n+0);

row = zeros(rows*2*x,1);
col = zeros(rows*2*x,1);
val = zeros(rows*2*x,1);
pos = +0;

for j = 1 : x^(n-1)
    if (j < x^(n-1))

    for k = max(1,x*(j-1)+1) : min(x^n,x*(j-1)+2*x)
        pos = pos+1;
        row(pos) = j;
        col(pos) = k;
        val(pos) = w2(k-x*(j-1));
    end

    else

    for k=1:x^n
        if k<=x
            pos = pos+1;
            row(pos) = j;
            col(pos) = k;
            val(pos) = w2(k+x);
        elseif k>= max(1,x*(j-1)+1)
            pos = pos+1;
            row(pos) = j;
            col(pos) = k;
            val(pos) = w2(k-x*(j-1));
        end
    end
    end
end

row = row(1:pos);
col = col(1:pos);
val = val(1:pos);

C = sparse(row,col,val,rows,cols);

row = zeros(rows*2*x,1);
col = zeros(rows*2*x,1);
val = zeros(rows*2*x,1);
pos = +0;

for j = 1 : x^(n-1)
    if (j < x^(n-1))

    for k = max(1,x*(j-1)+1) : min(x^n,x*(j-1)+2*x)
        pos = pos+1;
        row(pos) = j;
        col(pos) = k;
        val(pos) = w3(k-x*(j-1));
    end

    else

    for k=1:x^n
        if k<=x
            pos = pos+1;
            row(pos) = j;
            col(pos) = k;
            val(pos) = w3(k+x);
        elseif k>= max(1,x*(j-1)+1)
            pos = pos+1;
            row(pos) = j;
            col(pos) = k;
            val(pos) = w3(k-x*(j-1));
        end
    end
    end
end

row = row(1:pos);
col = col(1:pos);
val = val(1:pos);

D = sparse(row,col,val,rows,cols);

W1=[A;B;C;D];
% combined, this is the transform matrix which is for bigger pictures
W1=full(W1);
save F:\temporaryfiles\45.txt W1 -ascii
for i=1:3
W1f(:,:,i) = W1*F(:,:,i)*W1';
end
```

```
figure,subplot(1,2,1),imshow(F),title('original')
subplot(1,2,2),imshow(W1f),title('MWT')
imwrite(W1f, 'F:\temporaryfiles\0001.bmp')
```

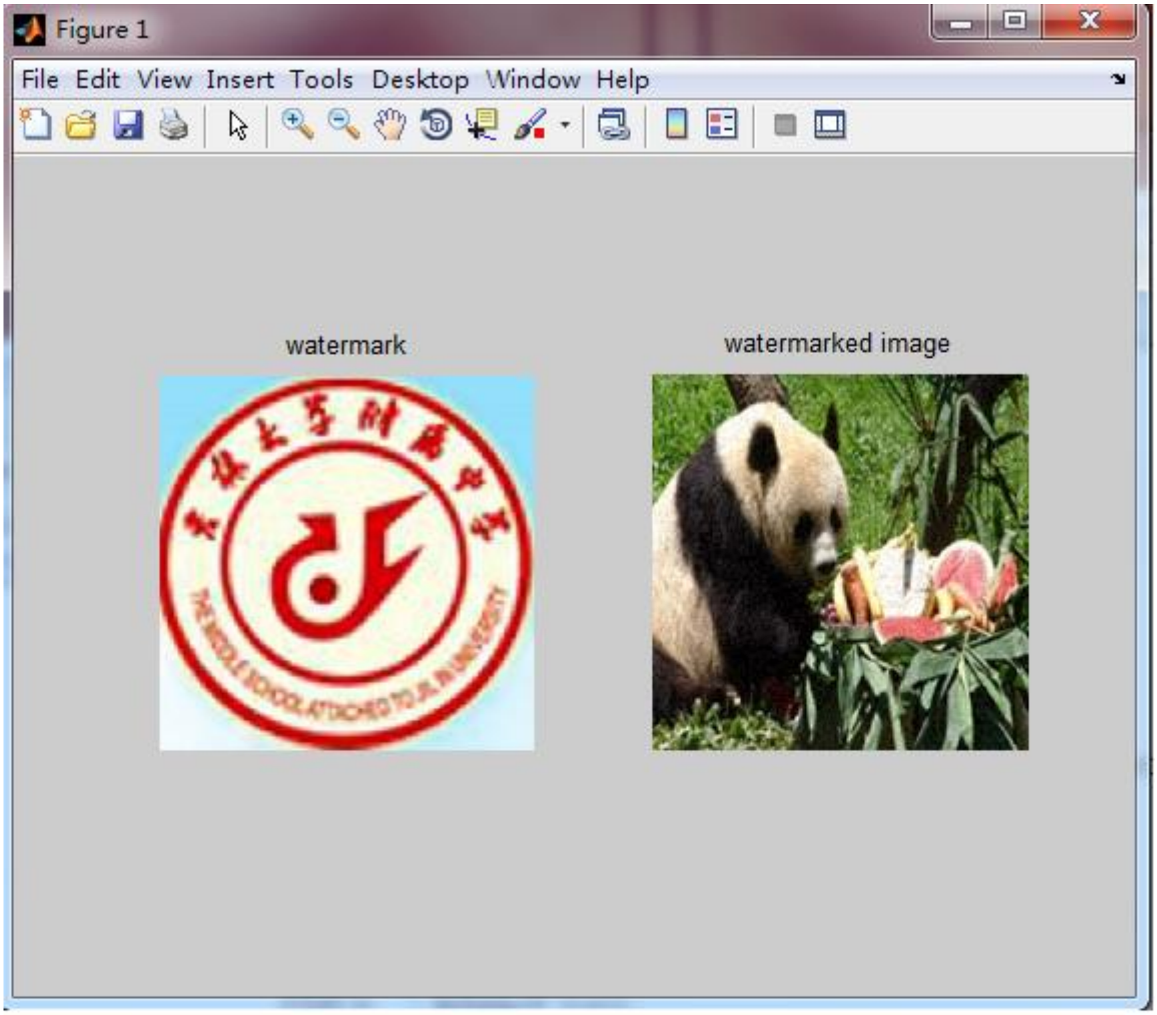


2) Watermark Embedding

```
clc;clear
Ix=imread('g:\pgm\a\Panda.bmp'); % picture original
Jx=imread('g:\pgm\a\logo256.bmp'); % watermark
T=textread('g:\pgm\a\4wt5.txt'); % the transform matrix we made

a=4;
n=4;
l=a^n; % size of the approximation
e=0.01; % epsilon, the embedding intensity indicator
```

```
P=rand(l); % pseudo-random numbers; to be replaced on quantum computers
I=zeros(l);
It=zeros(size(Ix));
J=zeros(l);
U1=zeros(1,3);U2=zeros(1,3);V1=zeros(1,3);V2=zeros(1,3);

% Watermark Embedding
% by calculating only approximation we achieve step 6
for t=1:3
I0=im2double(Ix(:,:,t));
J0=im2double(Jx(:,:,t));

% step 1 wavelet tramsform
I=T*I0*(T^(-1));
J=J0;
Ip=I;

% step 2 & 3 linear transform
U1(t)=max(max(I));U2(t)=min(min(I));
V1(t)=max(max(J));V2(t)=min(min(J));
I=(I+U1(t)-2*U2(t))*(pi/(6*(U1(t)-U2(t))));
J=(J+V1(t)-2*V2(t))*(pi/(6*(V1(t)-V2(t))));

% step 4 embed
    for i=1:l
    for j=1:l
        if P(i,j)>1/2
            I(i,j)=acos(cos(I(i,j))+e*cos(J(i,j)));
        else
            I(i,j)=asin(sin(I(i,j))+e*sin(J(i,j)));
        end
    end
    end

% step 5 % 6 inverse linear and wavelet transform
I=(I*6*(U1(t)-U2(t))/pi)+(2*U2(t)-U1(t));
```

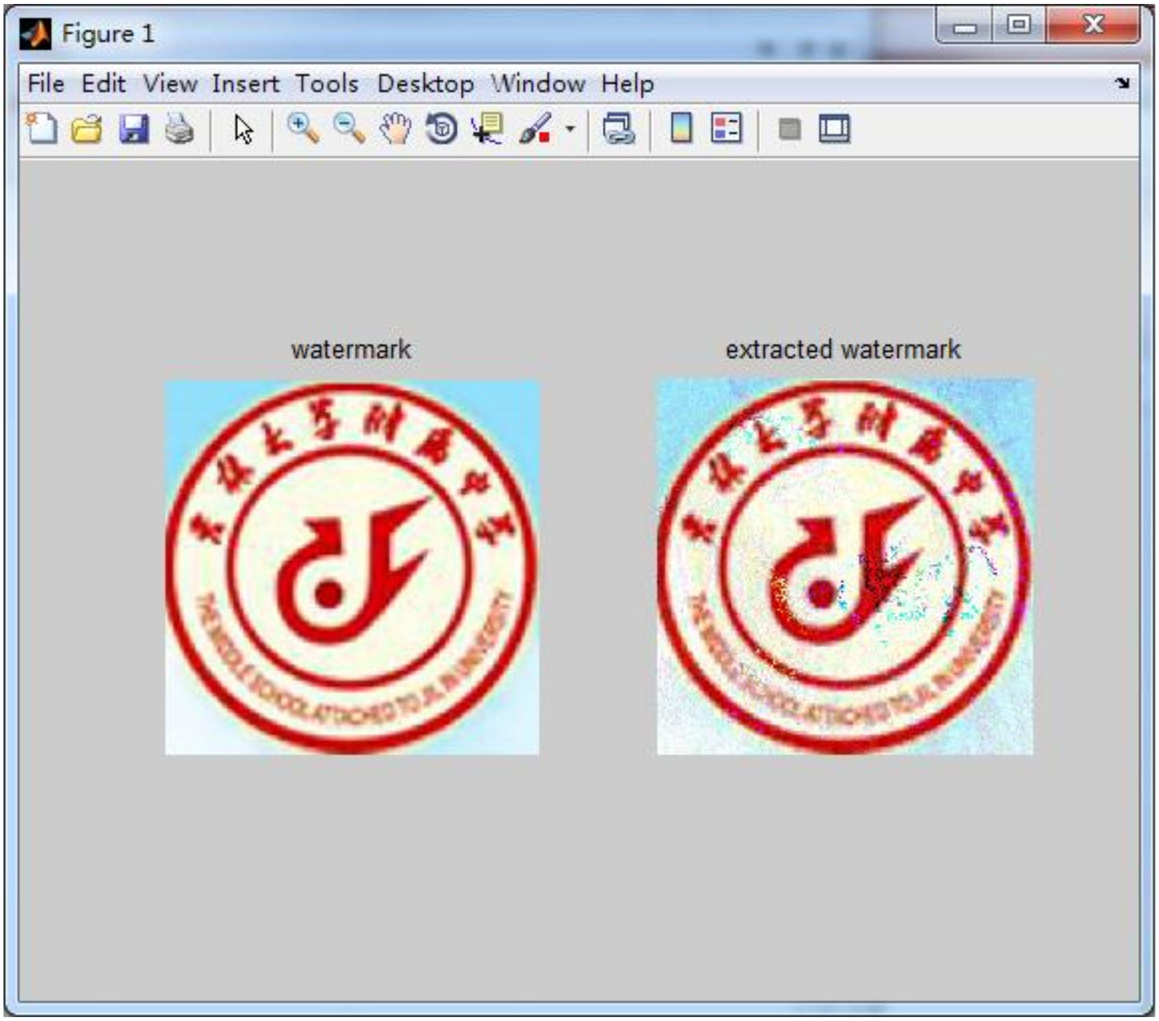
```
It(:,:,t)=(T^(-1))*I*T;
End

Ie=im2uint8(It);figure,imshow(Ie)
N=[U1;U2;V1;V2];
imwrite(I,'g:\temporaryfiles\result.bmp')
subplot(1,2,1),imshow(Jx),title('watermark')
subplot(1,2,2),imshow(Ie),title('watermarked image')
save g:\temporaryfiles\codebook4.txt P -ascii
save g:\temporaryfiles\max4.txt N –ascii
```



3) Watermark Extracting

```
clc;clear
P=textread('g:\pgm\a\codebook.txt'); % codebook
```

```
Z=textread('g:\pgm\a\max.txt'); % linear transform coefficient
Ix=imread('g:\pgm\a\Panda.bmp'); % original image
Ie=imread('g:\pgm\a\Panda01.bmp'); % embedded image
% for attacked watermark extract change this path
T=textread('g:\pgm\a\4wt5.txt'); % wavelet transform matrix

a=4;
n=4;
l=a^n; % picture size
e=0.01; % the indicator
It=zeros(l,l,3);
U1=Z(1,:);U2=Z(2,:);V1=Z(3,:);V2=Z(4,:);

% Watermark Data Extraction
for t=1:3
I0=im2double(Ix(:,:,t));
Ip=im2double(Ie(:,:,t));

% step 1 & 2 wavelet & linear transform
I=T*I0*(T^(-1));
J=T*Ip*(T^(-1));
W=zeros(l);
c=zeros(l);
I=(I+U1(t)-2*U2(t))*(pi/(6*(U1(t)-U2(t))));
J=(J+U1(t)-2*U2(t))*(pi/(6*(U1(t)-U2(t))));

% step 3 watermark extract
for i=1:l
    for j=1:l
        if P(i,j)>1/2
            c(i,j)=(cos(J(i,j))-cos(I(i,j)))/e;
            W(i,j)=acos(c(i,j));
        else
             c(i,j)=(sin(J(i,j))-sin(I(i,j)))/e;
            W(i,j)=asin(c(i,j));
        end
    end
end

% step 4 linear transform
W=(W*6*(V1(t)-V2(t))/pi)+(2*V2(t)-V1(t));
It(:,:,t)=W;
%disp(It)
end

Ie=im2uint8(It);
%imwrite(Ie,'g:\pgm\a\origin\watermarkgot.bmp')
imshow(Ie),title('extracted watermark')
imwrite(Ie,'g:\temporaryfiles\wtmk0.bmp')
```



4) Attacking & Calculating PSNR and RS

```
clc;clear
I=imread('g:\pgm\a\Panda01.bmp'); % original picture
I1=imnoise(I,'gaussian',0,0.002);
I2=imnoise(I,'salt & pepper',0.001);
I3=imnoise(I,'speckle',0.01);
I4=imresize(imresize(I,0.25),4); % attack pictures
% coefficients can be changed to make different attack effects

subplot(2,2,1),imshow(I1),title('gaussian')
subplot(2,2,2),imshow(I2),title('salt&pepper')
subplot(2,2,3),imshow(I3),title('speckle')
subplot(2,2,4),imshow(I4),title('compression')
imwrite(I1,'g:\pgm\a\gau.bmp');
imwrite(I2,'g:\pgm\a\s&p.bmp');
imwrite(I3,'g:\pgm\a\spk.bmp');
imwrite(I4,'g:\pgm\a\com.bmp');

% MSE calculating
MSE=0;
for u=1:3
    f1=double(I(:,:,u));
    f2=double(I3(:,:,u));
    % change the number to calculate different PSNRs
    [m,n]=size(f1);
    for i=1:m
        for j=1:n
        MSE=MSE+(f2(i,j)-f1(i,j))^2;
        end
    end
end
MSE=MSE/(m*n*3);
PSNR=10*log10(255^2/MSE);disp(PSNR)
```

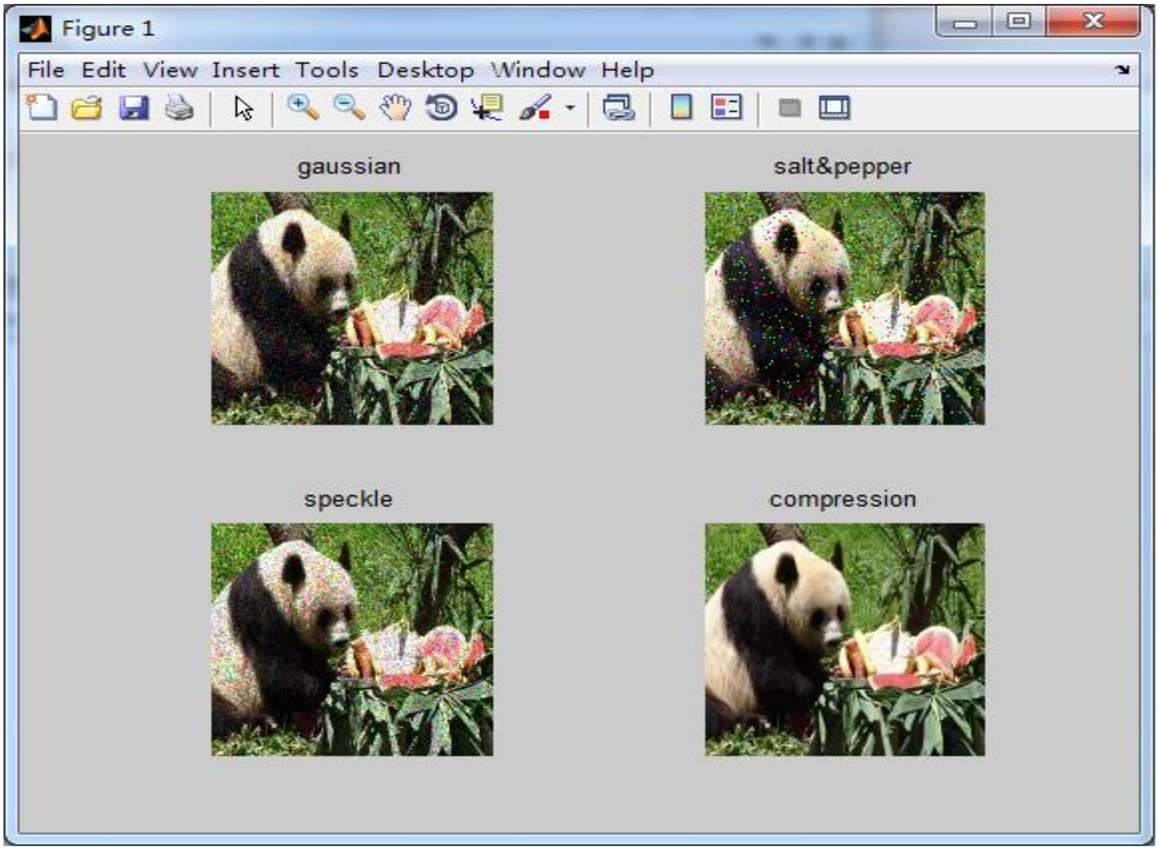


Figure 1
File Edit View Insert Tools Desktop Window Help
gaussian
salt&pepper
speckle
compression

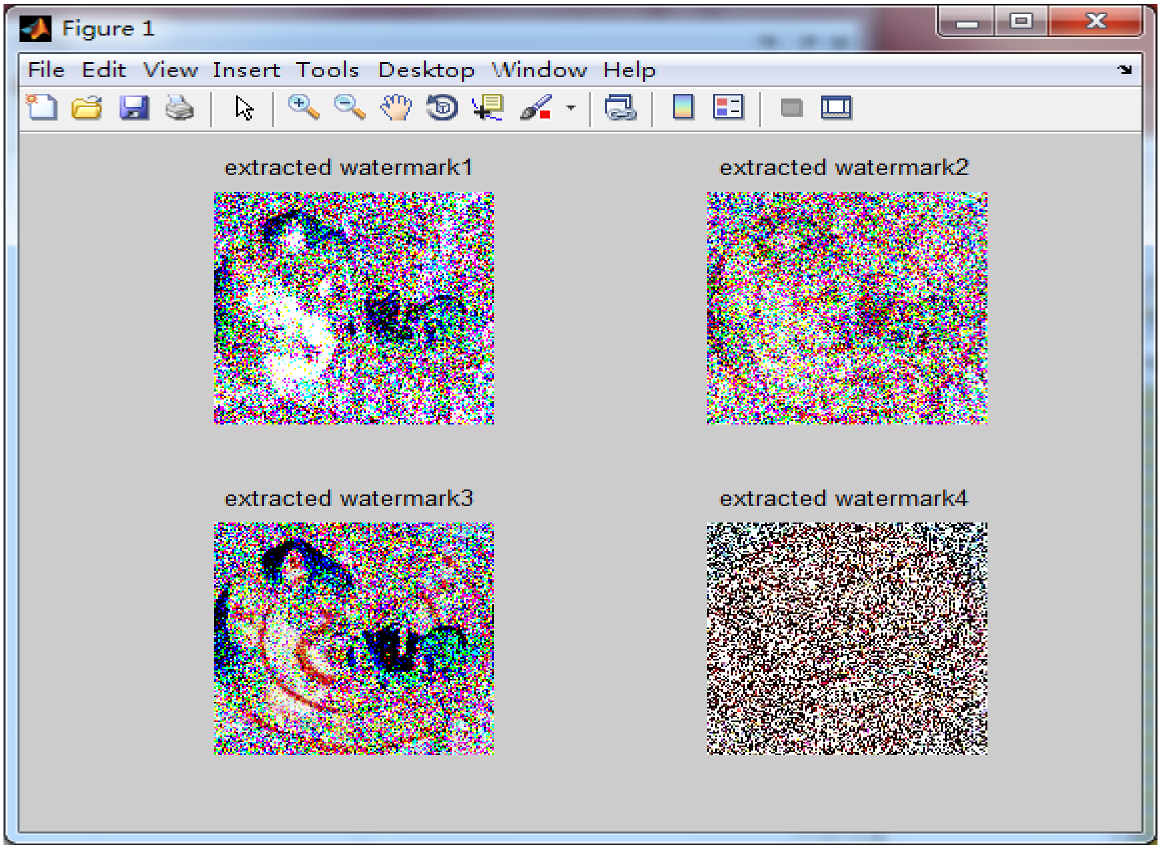


Figure 1
File Edit View Insert Tools Desktop Window Help
extracted watermark1
extracted watermark2
extracted watermark3
extracted watermark4

5) Calculating Relative Similarity (RS)

```
clc;clear
I=imread('g:\pgm\a\Panda.bmp'); % original picture
J=imread('g:\pgm\a\Panda01.bmp'); % watermarked picture
RS=1-(max(sum(sum((I-J))))/max(sum(sum((J)))));
disp(RS)
```

Appendix B

Resumes

Tong Liu (1998-) is with the Affiliated High School of Jilin University, 11th Grader now.

Awards:

The gold medal on 9th IMC(International Mathematics Contest), October 2008.

The first place on 14th HuaLuogeng Gold Cup Young Mathematicians Invitational Contest (National Final), April, 2009.

The first place of Outstanding Achievement and Honor Roll of Exceptional Performance on25thAMC8, Nov., 2009.

The first place of achievement on 11th AMC 10, Feb., 2010.

The gold medal on 21st Hope Cup Mathematics Contest, April, 2010.

The silver medal on 3rdHuaLuogeng Gold Cup Young Mathematicians Invitational Contest for China, Hong Kong, Macao, and Taiwan, August, 2010.

The gold medal on 15thHuaLuogeng Gold Cup Young Mathematicians Invitational Contest(National final), Oct., 2010.

The gold medal on 1st WMTC, intermediate group(World Mathematics Team Championship), Nov., 2010.

The first place of achievement on 63rd AMC 12 and the first place on 30th AIME, Feb., 2012

The first prize on the China's National Senior High School Mathematics Competition, Oct., 2012

The Honorable Mention of 15th High School Mathematical Contest in Modeling, Nov.,2012

The first prize on the China's National Senior High School Mathematics Competition, Oct., 2013 with a qualification of 29th China Mathematics Olympic

The second prize on the China's National Senior High School Physics Competition, Sept., 2013

The gold medal on 4th WMTC, advanced group, Nov.,2013

Extracurricular Activities:

Joined in the MathPath summer camp in USA, July, 2011.

Attended the JMM(American Joint Mathematics Meetings) and participated "undergraduate research poster session" with the joint research "M-band Wavelet and Cosine Transform Based Watermarking Algorithm Using Principal Component Analysis", Jan., 2012.

Presented in the WASET Conferences in Paris, with the joint research "M-band Wavelet and Cosine Transform Based Watermarking Algorithm Using Principal Component Analysis". Dec., 2012.

Published the joint research "M-band Wavelet and Cosine Transform Based Watermarking Algorithm Using Randomization and Principal Component Analysis" in International Journal of Science and Engineering Investigations, Feb., 2013.

Xuan Xu (1998-) is with the Affiliated High School to Jilin University, 10th Grader now.

Awards:

The bronze medal on 21st Hope Cup Mathematics Contest, Apr., 2010.

The silver medal on 15thHuaLuogengGold Cup Young Mathematicians Invitational Contest(National final), Oct., 2010.

The first prize on the China's National Middle School Mathematics Competition, April, 2011.

Gold medal on 2nd WMTC(World Mathematics Team Championship), Nov., 2011.

The perfect score: 25/25 and the first place of Outstanding Gold Medal Achievement on 25th AMC8, Nov., 2011.

The first place of achievement and Distinction Honor Roll on29th AMC 10 (scored 145.5/150), Feb., 2012.

The third prize on the China's National Senior High School Mathematics Competition, Oct., 2012.

The Honorable Mention of 15th High School Mathematical Contest in Modeling, Nov.,2012

Honor Roll on 64th AMC 12, Feb., 2013.

The second prize on the China's National Senior High School Mathematics Competition, Oct., 2013.

Extracurricular Activities:

Joined in the MathPath summer camp in USA, July, 2011.

Attended the JMM(American Joint Mathematics Meetings) and participated "undergraduate research poster session" with the joint research "M-band Wavelet and Cosine Transform Based Watermarking Algorithm Using Principal Component Analysis", Jan., 2012.

Presented in the WASET Conference in Paris, with the joint research "M-band Wavelet and Cosine Transform Based Watermarking Algorithm Using Principal Component Analysis". Dec., 2012.

Published the joint research "M-band Wavelet and Cosine Transform Based Watermarking Algorithm Using Randomization and Principal Component Analysis" in International Journal of Science and Engineering Investigations, Feb., 2013.