Available online at www.sciencedirect.com
ScienceDirect

Computer Standards & Interfaces 30 (2008) 271 – 287

COMPUTER STANDARDS & INTERFACES

www.elsevier.com/locate/csi

Content adaptive watermarking for multimedia fingerprinting

Yu-Tzu Lin [a,*], Ja-Ling Wu [a,b]

[a] *Communications and Multimedia Laboratory, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan*
[b] *Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan*

Received 30 July 2006; received in revised form 15 July 2007; accepted 10 August 2007
Available online 29 August 2007

Abstract

This paper presents a feasible system for multimedia fingerprinting. One of the important problems of fingerprinting is watermarking strategies for the fingerprints, in other words, where to embed and how to embed. We address these two major problems of fingerprint-watermarking for multimedia in the proposed content-adaptive watermarking scheme. First, the fingerprint codes are often very long if the applicable resisting ability to colluders and large customer bases are needed. As a result, the acute degradation of fingerprinted content may be unacceptable. We design a strength-decision algorithm, on the basis of neural networks, to adaptively embed the long-length fingerprints with suitable magnitudes to different image regions. This adaptive watermarking technique maintains the equilibrium of the robustness and the imperceptibility without the effort to deal with visual models. Second, we analyze the disturbance of collusion attacks on images and propose an optimization algorithm which can select better embedding positions to resist collusion attacks and preserve acceptable transparency of the watermark according to different multimedia contents. In addition, we consider the lossy property of multimedia watermarking and use a sequential detection strategy to identify colluders, which can tolerate erasures and errors possibly induced in the watermarking process or communication channel. Experimental results show the high detection correctness of traitor tracing. It implies that our fingerprinting system, constructed by applying c-TA code to the content-adaptive watermarking scheme and a sequential detection algorithm, is effective for multimedia application. One can replace the fingerprint codes in our system with other existing codes to obtain effective fingerprinting systems with higher tracing correctness and practical parameters (reasonable collusion-resilience and applicable size of customer bases).

Keywords: Fingerprinting; Watermarking; Traitor tracing; Collusion-resistance

1. Introduction

In applications of multimedia distribution, such as pay-TV scenario, data authorized to some privileged users are provided against unprivileged ones. A fingerprinting system protects media data by embedding a unique key to the host data of each user. These keys can be used to identify the source of illegal copies. However, a coalition of users may collude to make an illegal copy, which is different from all colluders' copies. Collusion-resistant fingerprinting devotes to solve this problem. Boneh and Shaw initiated the collusion-secure fingerprinting [1], which considers how to design the fingerprint codes to resist the coalition of c traitors, called frameproof codes. [2] proposes a traceability (TA) scheme and describes the resilient schemes and threshold tracing schemes, which inherit from the ideas of broadcast encryption schemes [3]. Stinson presented several constructions [4,5] for broadcast encryption schemes, such as orthogonal arrays, universal hash families and t-designs. Combinatorial properties of frameproof codes and traceability codes then are derived in [6], in which a c-TA scheme is defined and the combinatorial structures like t-designs and packing designs are used to construct frameproof codes and traceability schemes. Besides theoretical works from the viewpoint of fingerprint design, some literatures proposed practical fingerprinting systems for multimedia. In [7], a collusion-secure fingerprinting scheme based on finite geometries is presented. [8] constructed an anti-collusion code for multimedia by using Balanced Incomplete Block Design (BIBD) [9], and applied the code-modulation technique to reduce the spreading sequences significantly. This practical and complete fingerprinting scheme provides a good research direction in fingerprinting. In addition to the combinatorial methods, another trend of

* Corresponding author. Tel.: +886 916 656500; fax: +886 2 23628167.
E-mail address: linyt@cmlab.csie.ntu.edu.tw (Y.-T. Lin).

0920-5489/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2007.08.012

fingerprint codes employs the concept of error-correcting codes (ECC). The idea is that one can identify the colluders' codewords if the minimum distance of the code is large enough. In [10], Reed–Solomon codes are used to construct a traceable code and list decoding techniques are used to find all possible pirate coalitions efficiently. The authors also described how coding-theoretic techniques may be applied to the traitor tracing problem for linear codes.

Several researches [11] construct the collusion-secure fingerprints based on orthogonal noise-like signals. However, there is no systematic decoding algorithm for these methods, and it is not easy to find so many orthogonal signals. [8] combined the orthogonal noise-link signals and combinatorial designed codes using CDMA to produce fingerprints. The combinatorial design can help the finding of structured fingerprinting codes, which provide good properties for uniquely identifying colluders. But the combinatorial design is not trivial, thus the construction of this type of codes is difficult and so the scalability is limited. The ECC-based fingerprinting codes treat the fingerprint-generation problem as an ECC design problem. The mathematical operations on finite fields for ECCs, such as Reed–Solomon codes, provide more regularity and efficiency on both encoding and decoding than that of the combinatorial methods. However, the possibility of analyzing the fingerprinting though the model of communication channels makes the fingerprints more rigid to attacks. However, the attempt to increase the minimum distance between codewords results in an abrupt increasing of the code-length. In this paper, we choose ECC-based codes to construct our fingerprint code because of the systematicness of ECC. Besides the inherent property of large minimum distances, the demand for large size of customer base and enough resilience-size c (the theoretical maximum number of colluders resisted in the fingerprinting system) in practical applications still makes the code-length of ECC-based fingerprinting codes very long. The code-length of the c-TA code proposed in [12] is even much longer in order to obtain the "sequential tracing" [13] property and make the number of codewords independent of the resilience-size c. We use this code and consider how to embed the long-length code to images without influencing the imperceptibility significantly.

Digital fingerprinting often embeds the key to the protected data using watermarking techniques. Most of the watermarking schemes are developed to fit the following requirements: robustness, high capacity, and fidelity. However, a trade-off should be considered among these three requirements. The scenario is apparent in fingerprinting problems because of the long-length feature mentioned above. Multimedia data are not so sensitive to disturbance in data values as compared with other digital data, this implies that multimedia data can tolerate more embedding bits. Furthermore, Human Visual System (HVS) has been introduced in some watermarking schemes [14–16] to improve the quality of watermarked images by embedding watermark with maximum strength based on visual models, so as to reach the equilibrium of the prescribed three watermarking requirements. However, the construction of HVS model involves formulating the estimation of the image quality, which is still an open problem. Therefore, some researchers apply neural networks to decide the maximum watermarking strength automatically. [17] uses the back-propagation network to compute the maximally tolerable watermark strength of an image and takes the first 4096 discrete wavelet transform coefficients as the input of the network. But in the training process, user may be baffled when assigning the watermarking strength to the whole image and the assignments may be biased because this method ignores the variety of different regions in an image. [18] feds the DCT coefficients, in one 8×8 block, to the neural network and obtains the embedding strength of this block. But the decision, in the training process, to find appropriate embedding values of an 8×8 block for humans is difficult. In our work, the neural network is used to learn the suitable watermarking strength for various image regions, and the learning is unsupervised.

Collusion attacks are critical to the fingerprinting problem. A group of legal users may collude by combining their fingerprints to create pirate copies and spread them. These traitors are intent to destroy their unique identification and let the arbiter hard to trace them. Collusion-resilient fingerprinting schemes are developed to resist the colluders' malice. The fingerprint-designing mentioned above tries to construct codes resistant to collusion attacks, nevertheless, poor watermarking strategies may weaken the resistance of fingerprinting schemes. In [7], the position sequences for embedding watermarks are generated randomly from the user key as a seed, so as to avoid the fingerprint-bits being eliminated by collusion attacks. [19] proposes a video fingerprinting architecture by combining fingerprinting and decryption based on partial encryption. There are two types of encryption keys: one is served as pointers to subsets of coefficients to encrypt and another is the scrambling key dictating the order in which the coefficients are permuted within a given subset. Instead of building the anti-collusion scheme by randomness or encryption, we extend the idea of watermark optimization [20] and devise an algorithm to select suitable embedding positions for fingerprints in order to intensify the collusion-resistance.

Another challenge in traitor tracing scheme is the detection of traitors. When the arbiter gets one pirated version of the protected content, he should extract the counterfeit fingerprint from the content and analyze it so as to find the colluders. [10] uses the list decoding techniques of ECC to efficiently find all possible coalitions. In [8], a tree-structured detection is proposed for the orthogonally modulated fingerprints and a soft-thresholding detection algorithm is used to decide the colluders adaptively [21] applies soft-decision list decoding to enhance the traceability of Reed–Solomon codes, which can list the colluders iteratively by computing the reliability matrix. Because our watermarking scheme is lossy (even it is not lossy, errors may be introduced by many possible attacks in the communication channel), the hard-decision strategy for traitor detection is not applicable. We propose a sequential detection strategy for traitor tracing to identify possible colluders using the proposed membership function.

Few literatures considered practical parameters for actual applications. In this paper, we try to find a practical multimedia fingerprinting scheme to take both the need of large number of fingerprints for numerous users and the characteristics of multimedia data into account. This multimedia fingerprinting

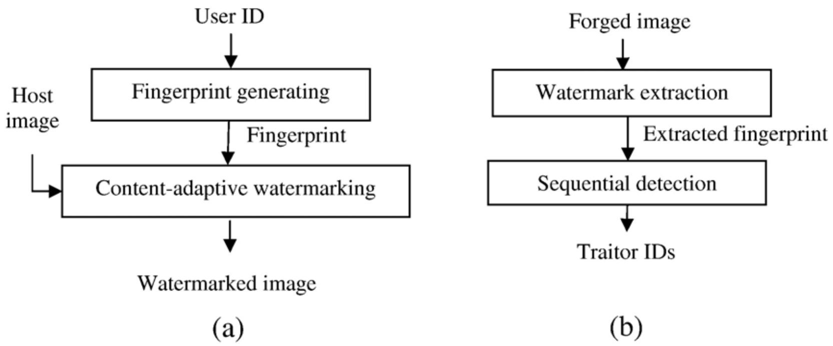


Fig. 1. System overview: (a) fingerprinting scheme, and (b) traitor tracing scheme.

scheme embeds a sequential c-TA code, based on [12], into appropriately selected positions with the strength computed from a watermark-strength decision technique to make the code applicable to multimedia, and then a sequential traitor detection algorithm which can tolerate erasures and errors is used to identify at least one of the traitors. The proposed content-adaptive watermarking scheme uses Self-Organizing Maps (SOMs) [22] to tune the watermark strength locally and adaptively to ensure the embedded watermark perceptually invisible and robust. The derived position-selection algorithm will selects appropriate embedding positions by first choosing a suitable energy function which represents the degree of collusion-resilience, then changing the selected positions iteratively to approach high energy and finally find the optimal positions.

In the rest of this paper, we will first briefly describe our system in Section 2. In Section 3, we present the code used for traitor tracing. Details of content-adaptive watermarking strategies are provided in Section 4. Section 5 states the algorithm of the sequential detection for traitor tracing. Section 6 presents our experiments and Section 7 gives the concluding remarks.

2. System overview

Fig. 1 shows our system architecture. Fig. 1(a) illustrates the fingerprinting process, in which fingerprints are generated according to user IDs and embedded into the host image using an adaptive image watermarking technique, and the produced watermarked images are then assigned to users individually. In the fingerprint-generation step, a c-TA code is constructed from ECCs that satisfies a bound on their minimum distance. The content-adaptive watermarking scheme determines the maximum watermarking strength locally by the SOM neuron network using Fourier transform coefficients and picks suitable embedding positions from the candidate positions for the purpose of resisting collusion attacks (will be detailed in Section 4). After watermark embedding, each user acquires a particular watermarked copy with his particular fingerprint in it.

If some legal users collude to produce the pirate image, at least one of the colluders should be identified. We show the traitor tracing scheme in Fig. 1(b). First, the pirate code is extracted from the forged image in the watermark extraction stage. The traitors can be identified according to the extracted code. Then we find at least one of the colluders by using the sequential detection strategy (will be detailed in Section 5). For the ease of understanding, in the next section, the traceability code used in our system is briefly reviewed first.

3. Traceability code

A *c*-traceability scheme is defined in [6] and some related works [10,23] discuss the codes with the property of *c*-traceability, which are called the *c*-traceability codes.

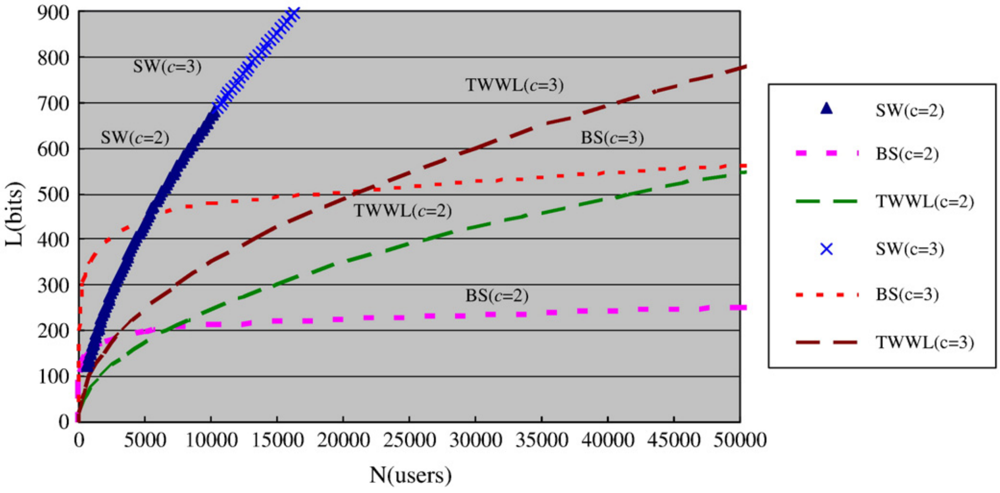


Fig. 2. The required code-length to serve certain number of users.

Definition 1. A q-ary ECC C having N codewords of length L is called an $(L, N, D)_q$-ECC if the minimum distance between any two codewords of C is D.

Definition 2. [23] Suppose C is an $(L, N, D)_q$-ECC and $c \geq 2$ is an integer. Let $C_i \subseteq C$, $i=1,2,\ldots, t$, be all the subsets of C such that $|C_i| \leq c$, then C is a c-TA code if for all i and all $x \in desc(C_i)$, there is at least one codeword $y \in C_i$ such that $|I(x, y)| > |I(x, z)|$ for any $z \in C / C_i$ (in which $I(x, y) = \{I\colon x_i = y_i\}$ and $desc(C_0) = \{x \in Q^N\colon x_i \in \{a_i\colon a \in C_0\}, 1 \leq I \leq N\}$ where Q is the alphabet).

The c-TA code can be constructed by various methods [1,10,12]. [22] states that we can find a c-TA code by looking for the $(L, N, D)_q$-ECC with large minimum distance D.

Theorem 1. [23] Suppose that C is an $(L, N, D)_q$-ECC having minimum Hamming distance

$$D > \left(1 - 1/c^2\right)L \tag{1}$$

then C is a c-TA code.

Our traitor tracing scheme is based on the sequential c-traceability code [13] proposed in [12], which will be detailed in Section 3.1.

3.1. c-TA code construction

The adopted c-TA code is constructed from finite elements in GF(q^k). The codeword is the vector

$$\left(\mathrm{Tr}\left(\alpha x_1^s + \beta\right), \mathrm{Tr}\left(\alpha x_2^s + \beta\right), \ldots, \mathrm{Tr}\left(\alpha x_{q^k}^s + \beta\right)\right) \tag{2}$$

where Tr(·) is the tracing function $\left(\mathrm{Tr}(\omega) = \omega + \omega^q + \omega^{q^2} + \ldots + \omega^{q^{k-1}}\right)$, $\alpha \in \mathrm{GF}(q^k)*$, and $\beta \in \{\beta_1, \beta_2, \ldots, \beta_\alpha\}$ be q elements of GF(q^k) whose trace values are pairwise distinct. The positive integer s satisfies $k=2t$, $s < q^{k/2+1}$, and $\exists r$ such that $r|t$, $q^r = -1$ (mod s). If

$$c < \frac{-1 + \sqrt{1 + 4q^{3k/2}\left(q^{k/2} + (s-1)(q-1)\right)}}{2q^{k/2-1}\left(q^{k/2} + (s-1)(q-1)\right)} \tag{3}$$

then one can show that the corresponding minimum distance D of the derived code satisfies the criterion (1) of Theorem 1, which implies a c-TA code is obtained. For example, if q is 32 and k is 2, we can set t and s to 1 and 3, respectively, thus the codeword length L is 5120, the codeword size N is 32736, and the resilience-size c is 2.

The obtained c-TA code vectors are then used as our fingerprints, different user IDs are mapped to different parameters (α, β, s), and so different codewords [12].

3.2. Analysis of the c-TA code

Consider the codeword vector in (2), we can derive that $L = q^k$ and $N = q(q^k - 1)$. When q is large enough, the resilience-size c is approximately equal to $\sqrt[4]{q}$, which is independent of k. So for a fixed q, we can increase N without influencing the level of security c. This property makes our code construction flexible in the number of users. But for Reed–Solomon codes, c approximately equals to $\sqrt{(q-1)(k-1)}$. So the proposed c-TA code improves the drawback of Reed–Solomon codes in which increasing k will cause the reduction of c.

Fig. 2 demonstrates the comparison of the fingerprint codes proposed in [1], [8], and [12] (which are represented as BS, TWWL, and SW, respectively) on L, N, and c. It is only an analysis of the growing trend on L and N, but not guaranteed that the corresponding construction-methods of c-TA code for points lied on these curves all exist. The curves of the method proposed in [12] are the steepest, as shown in Fig. 2, which means the produced code must increase its length most

Table 1
Some examples of parameters for the c-TA code proposed in [11]

q	k	c	L	N	q	k	c	L	N
22	2	2	2420	10,626	70	2	2	34,300	342,930
23	2	2	2645	12,144	71	2	2	35,287	357,840
24	2	2	2880	13,800	72	2	2	36,288	373,176
25	2	2	3125	15,600	73	2	2	37,303	388,944
26	2	2	3380	17,550	74	2	2	38,332	405,150
27	2	2	3645	19,656	75	2	2	39,375	421,800
28	2	2	3920	21,924	76	2	2	40,432	438,900
29	2	2	4205	24,360	77	2	2	41,503	456,456
30	2	2	4500	26,970	78	2	2	42,588	474,474
31	2	2	4805	29,760	79	2	2	43,687	492,960
32	2	2	5120	32,736	80	2	2	44,800	511,920
33	2	2	6534	35,904	81	2	2	45,927	531,360
34	2	2	6936	39,270	82	2	2	47,068	551,286
35	2	2	7350	42,840	83	2	2	48,223	571,704
36	2	2	7776	46,620	84	2	2	49,392	592,620
37	2	2	8214	50,616	85	2	2	50,575	614,040
38	2	2	8664	54,834	86	2	2	51,772	635,970
39	2	2	9126	59,280	87	2	2	52,983	658,416
40	2	2	9600	63,960	22	3	2	53,240	234,234
41	2	2	10,086	68,880	88	2	2	54,208	681,384
42	2	2	10,584	74,046	89	2	2	55,447	704,880
43	2	2	11,094	79,464	90	2	2	56,700	728,910
44	2	2	11,616	85,140	91	2	2	57,967	753,480
45	2	2	12,150	91,080	92	2	2	59,248	778,596
46	2	2	12,696	97,290	93	2	2	60,543	804,264
47	2	2	13,254	103,776	23	3	2	60,835	279,818
48	2	2	13,824	110,544	94	2	2	61,852	830,490
49	2	2	14,406	117,600	95	2	2	63,175	857,280
50	2	2	15,000	124,950	96	2	2	64,512	884,640
51	2	2	15,606	132,600	97	2	3	65,863	912,576
52	2	2	16,224	140,556	98	2	3	67,228	941,094
53	2	2	16,854	148,824	99	2	3	68,607	970,200
54	2	2	17,496	157,410	24	3	2	69,120	331,752
55	2	2	18,150	166,320	100	2	3	70,000	999,900
56	2	2	18,816	175,560	101	2	3	71,407	1,030,200
57	2	2	19,494	185,136	102	2	3	72,828	1,061,106
58	2	2	20,184	195,054	103	2	3	74,263	1,092,624
59	2	2	20,886	205,320	104	2	3	75,712	1,124,760
60	2	2	21,600	215,940	105	2	3	77,175	1,157,520
61	2	2	22,326	226,920	25	3	2	78,125	390,600
62	2	2	23,064	238,266	106	2	3	78,652	1,190,910
63	2	2	23,814	249,984	107	2	3	80,143	1,224,936
64	2	2	24,576	262,080	108	2	3	81,648	1,259,604
65	2	2	29,575	274,560	109	2	3	83,167	1,294,920
66	2	2	30,492	287,430	110	2	3	84,700	1,330,890
67	2	2	31,423	300,696	111	2	3	86,247	1,367,520
68	2	2	32,368	314,364	112	2	3	87,808	1,404,816
69	2	2	33,327	328,440	26	3	2	87,880	456,950

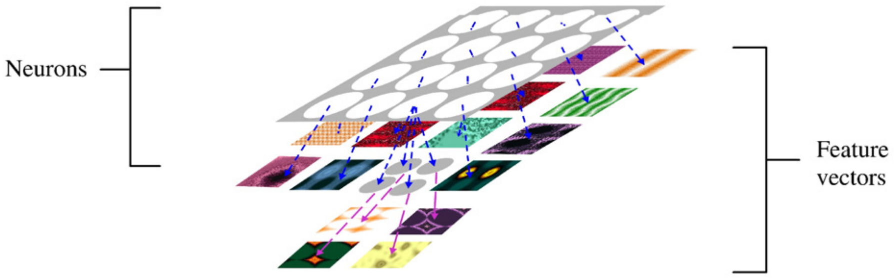


Fig. 3. The architecture of the adopted SOM network. Solid lines indicate the links between parent-nodes and child-nodes while dotted lines map neurons to feature vectors expressed in images.

significantly when the customer base becomes larger. In another words, we have to pay the cost of code-length to get the advantages of systematic code-construction and stableness of the resilience-size c when N is increased. Of course, how to get the optimal trade-off between the cost we paid and the advantage we got is one of our future research topics.

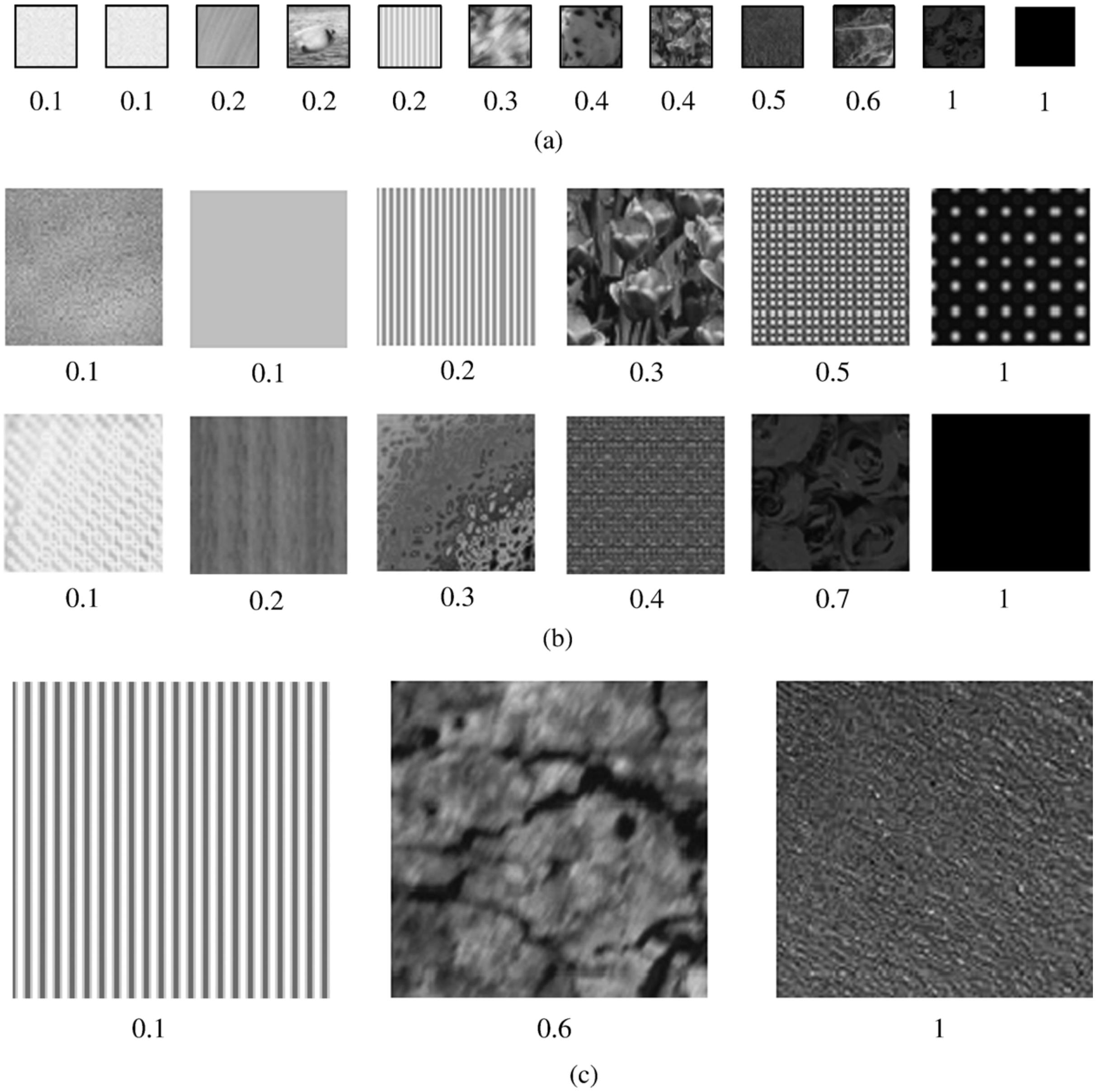


Fig. 4. Some of the training images of SOM, and values below the images are the corresponding JNDs (i.e. the maximum strength for imperceptible watermarking): (a) The sub-image size m is 32, (b) m is 64, and (c) m is 128.

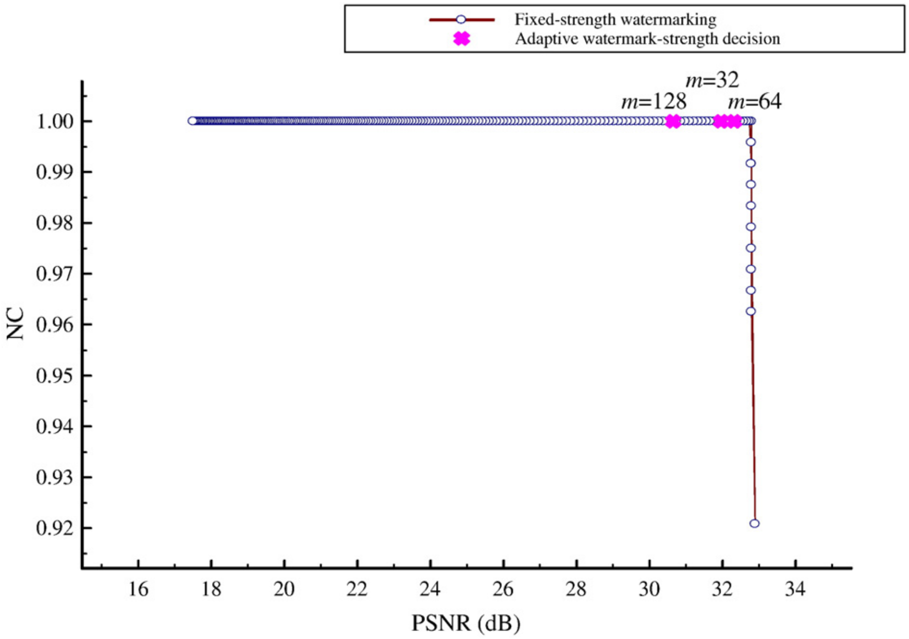


Fig. 5. The *NC-PSNR* curve of fixed-strength watermarking vs. results of adaptive watermark-strength decision, in which the size of test images is 256 × 256, the code-length of the fingerprints is 24,576, and sub-image size *m* is 32, 64, or 128.

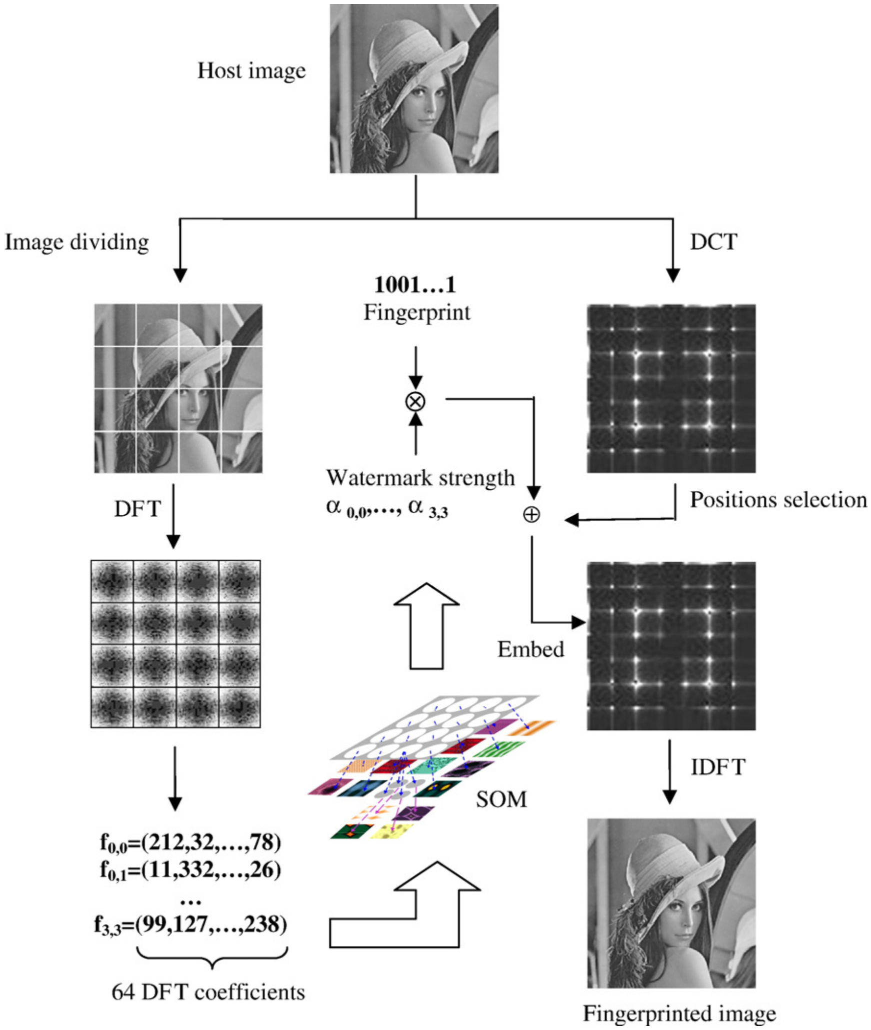
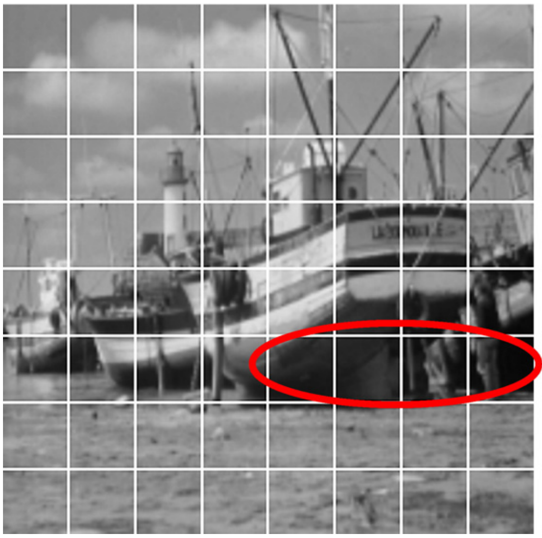


Fig. 6. The fingerprint embedding process.



0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
0.3	0.3	0.3	0.3	0.3	0.2	0.3	0.3
0.3	0.3	0.2	0.3	0.3	0.2	0.3	0.2
0.3	0.3	0.2	0.2	0.3	0.3	0.3	0.3
0.3	0.3	0.3	0.2	0.3	0.3	0.3	0.6
0.3	0.3	0.3	0.3	0.7	0.7	0.7	1
0.3	0.3	0.3	0.2	0.3	0.2	0.2	0.2
0.3	0.3	0.3	0.3	0.3	0.3	0.2	0.3

Fig. 7. The corresponding watermark-strength for each of the sub-images of the Fishingboat image.

3.3. Implementation of the c-TA code for multimedia

Table 1 gives some examples of parameters for the *c*-TA codes we used. In our experiments, the fingerprint codewords are constructed as in (2) from GF(64^2) with $s=5$ (we choose q as a power of 2 for the reason of binary operations), thus we can serve 262,080 users, and the length of the binary codewords is 24,576, which is much longer as compared with watermark-signals used in most of the watermarking techniques in the literatures, and therefore, embedding with these codewords may affect the watermark transparency significantly. The watermarking algorithm presented in the next section tries to solve this problem.

4. Content-adaptive watermarking

To achieve high imperceptibility while preserving robustness of the watermark and resistance to collusion attacks, we proposed a content-adaptive watermarking scheme. An SOM neural network, which can learn variant levels of tolerance of quality distortion for different multimedia contents, is built to

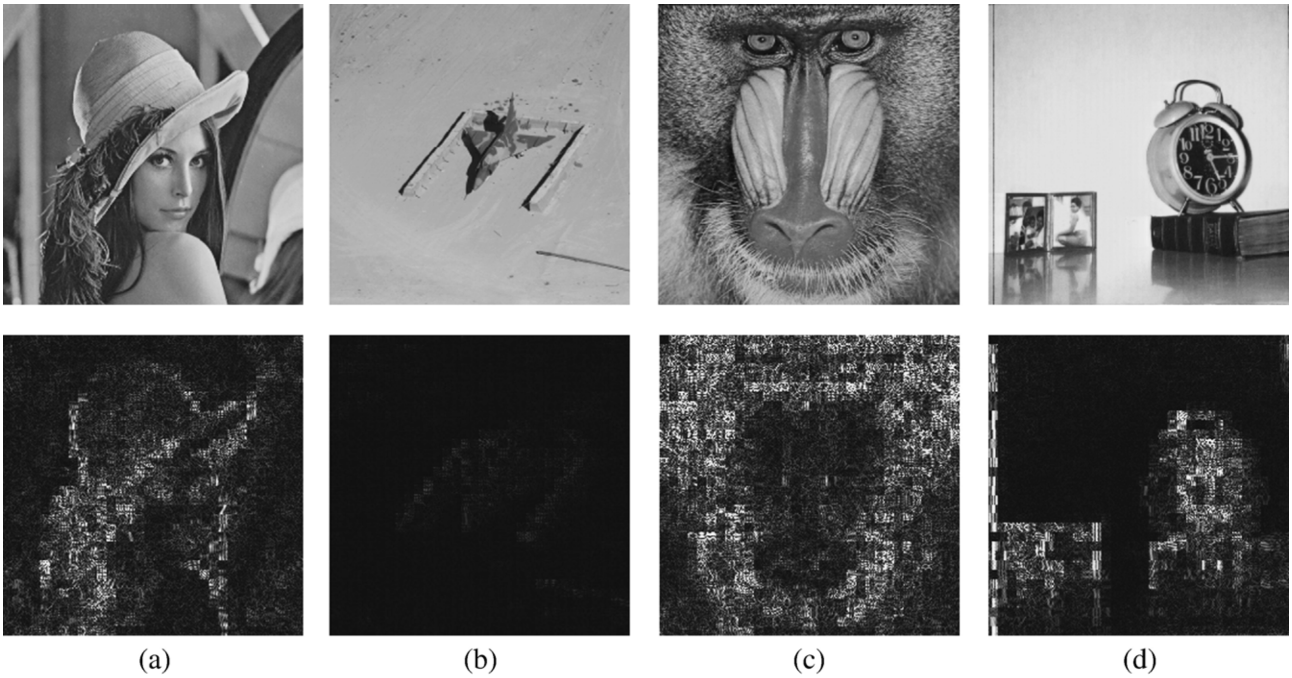
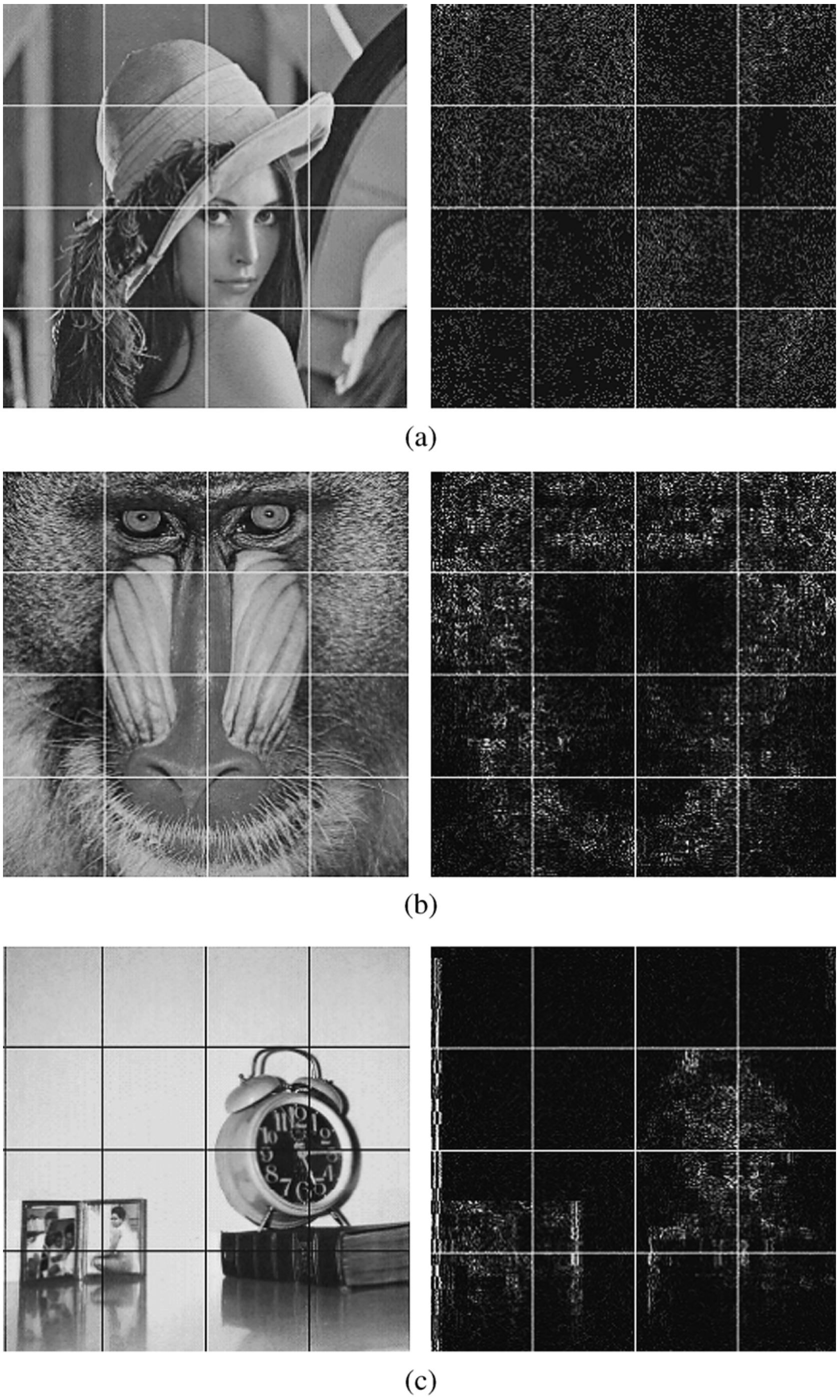


Fig. 8. The watermarks produced by adaptive watermark-strength decision for some example images: (a) Lena, (b) Airplane, (c) Baboon, and (d) Clock. (Top) Original images. (Bottom) Difference images for adaptive strength-decision watermarking, where "difference image" denotes the difference image between the original image and the watermarked image.



(a)

(b)

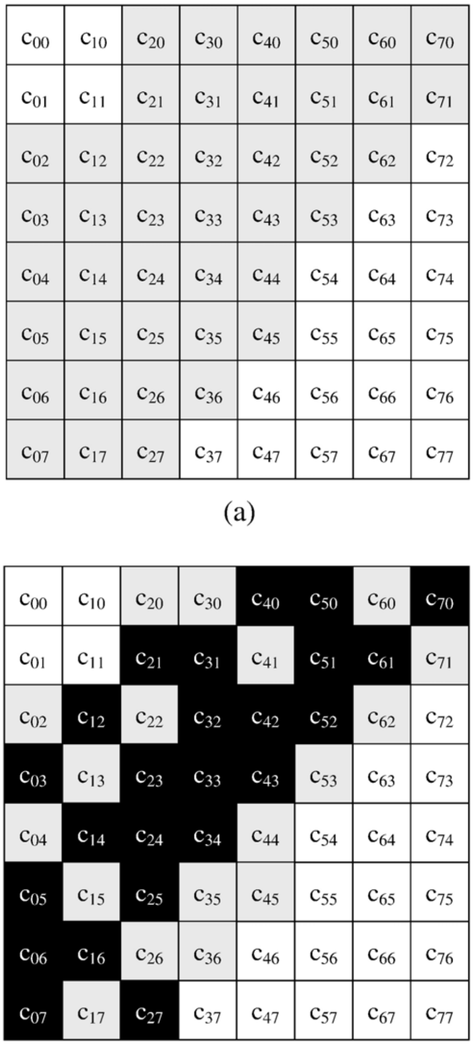
(c)

Fig. 9. Original images (left) and the difference images between watermarks of FSW and ADSW (right). The dividing lines illustrate the blocks for watermark-decision (where $m=64$): (a) Lena, (b) Baboon, and (c) Clock.

simulate the human visual system. After the SOM neural net is trained, it learns the ability to adjust the watermark strength adaptively according to the multimedia contents, so as to minimize the distortion of the content quality while preserving robustness. In this paper, we focus on image watermarking, and the proposed technology can be applied to other types of media in a similar way. The strength-decision mechanism will be detailed in the first part of this section and another issue of resistance to collusion attacks (i.e. where to embed) is considered and solved by an optimization procedure, will be described in the second part.

4.1. SOM architecture

The SOM neural network can imitate the mechanism of cerebral cortex. A 2-D map is used in our architecture to represent the cortex map memorizing the distortion sensitivity for variant image patterns. During the training period of the network, each


c00 c10 c20 c30 c40 c50 c60 c70
c01 c11 c21 c31 c41 c51 c61 c71
c02 c12 c22 c32 c42 c52 c62 c72
c03 c13 c23 c33 c43 c53 c63 c73
c04 c14 c24 c34 c44 c54 c64 c74
c05 c15 c25 c35 c45 c55 c65 c75
c06 c16 c26 c36 c46 c56 c66 c76
c07 c17 c27 c37 c47 c57 c67 c77

(a)

(b)

Fig. 10. The embedding positions: (a) Candidates of embedding DCT coefficients (shown in gray), and (b) One of the examples of positions selected for embedding (shown in black).

unit with a positive activity within the neighborhood of the winning unit updates its value toward the features of certain image appearances using Eq.(4), where w_i (t) is the weight vector of the ith unit in the tth iteration, $\alpha(t)$ is the learning rate, and N_c is the list of unit indices that make up the neighborhood.

$$w_i(t+1) = \begin{cases} w_i(t) + \alpha(t)(x - w_i(t)), i \in N_c \\ 0, \qquad\qquad\qquad\qquad \text{otherwise.} \end{cases} \tag{4}$$

After iterations of updating values of neurons by Kohonen's algorithm [22], nodes in the 2-D array organize themselves in response to the input vectors. The resulting map projects the input vectors onto a 2-D map preserving the natural topology of training data, which contains the feature vectors of various types of images, including smooth, high-frequency, regular-texture images, etc.

Initially, a 2-D SOM with 4 × 4 neurons is constructed. Each node is randomly assigned to feature values of one of the training samples. The learning algorithm performs the weight-update process using an approximation to the Mexican-hat function [24]. If one node is too coarse to represent certain category of training data, it is divided into 4 child-nodes. Further dividing should be proceeded if nodes after the first dividing are still too rough, and so on. The architecture of our SOM network is shown in Fig. 3. The feature vectors used in our SOM are obtained from performing the discrete Fourier transform (DFT) on the original data. Vectors derived from images with similar characteristics are trained to gather together in adjacent nodes. DFT coefficients are used as features because, in the Fourier domain, significant features are compressed into several coefficients only. Moreover, the good properties of scaling-invariance and rotation-invariance in the DFT domain will make our features approximate the human's impression closely.

In the training stage, training samples of size $m \times m$ images are first categorized manually. For each training image, we increase the watermark strength gradually until the watermark is perceptible, and then classify this watermarked image to that category in which all images can tolerate equal strength of the watermark, the corresponding strength is called the just

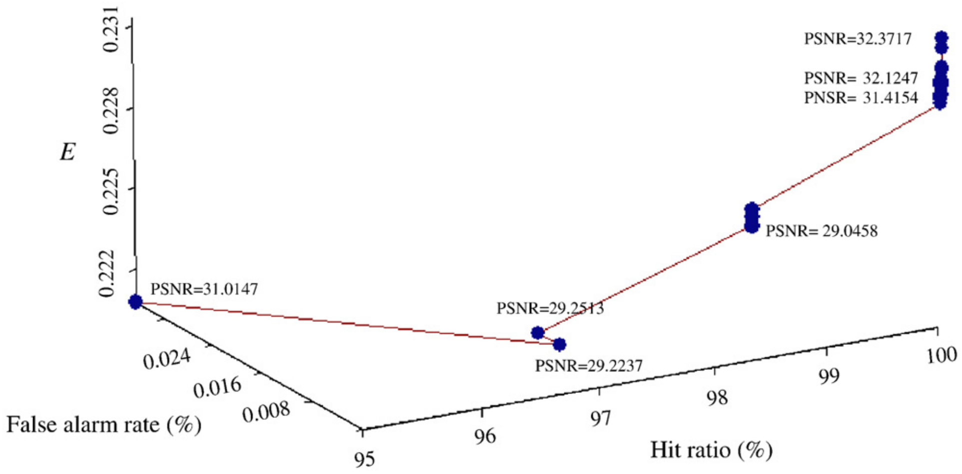


Fig. 11. The maximization process of the energy function E. The curve shows the variations of hit ratio, false alarm rate, and $PSNR$ in the maximization process.

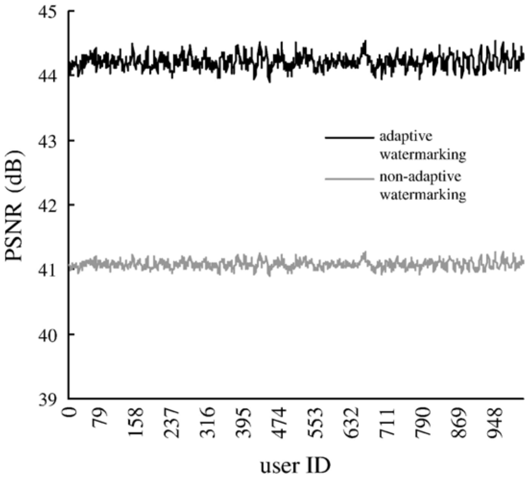


Fig. 12. Imperceptibility comparisons between the adaptive watermarking and the naïve watermarking schemes with fixed watermark strength.

noticeable distortion (JND) of that category and the value has been normalized to the closed interval [0,1]. In the network learning stage, we take the first 64 DFT coefficients of the classified training images as input vectors, which can be represented as a 64-dimension vector. If the feature vector of the current training sample is the nearest one to the center of certain category, the coordinate of that center is modified and moved toward the training vector. After iterations of learning, the SOM then eventually memories the maximum watermark strength imperceptible to humans for every category of the training images. In the recalling process, input image is divided into several $m \times m$ sub-images. These sub-images are then fed to the neural network as inputs. If features of the current input sub-image are most similar to features of certain category, the watermark is embedded into the sub-image according to the JND of this category. Some of the training samples of various sizes and their corresponding JNDs are shown in Fig. 4. Sizes of images will affect human's perception and may cause the decisions of JNDs distinct.

4.2. Adaptive watermark-strength decision

After fingerprint generation, the fingerprint codewords should be embedded into the host image. First, the host image is divided into several sub-images of size $m \times m$. Our experiments show that the performance is the best when m is 64 (see Fig. 5, which will be addressed later in this section). Second, DFT coefficients of these sub-images are fed to the SOM individually to get their watermark strengths. Finally, the fingerprint is added to DCT coefficients (we select suitable coefficients to embed using an optimization algorithm, which will be detailed in Section 4.3) for every 8×8 block of the host image according to the obtained strength in the previous stage. The watermark embedding procedure is illustrated in Fig. 6.

Our watermarking method is oblivious, in other words, the watermark detection does not need the original image. Elements of the codeword vector (2) are in GF(q), we first denote them in binary form $\left(v_0, v_1, \ldots, v_{q^k-1}\right)$, where $v_i \in \{0,1\}$. Then, modify the DCT coefficients block by block as follows.

$$\begin{cases} \widetilde{x}_{k,b} = x_{k,b-1} + \alpha_s \beta Q_k x_{0,b}, \text{if } v_i = 1 \\ \widetilde{x}_{k,b} = x_{k,b-1} - \alpha_s \beta Q_k x_{0,b}, \text{if } v_i = 0 \end{cases}, \tag{5}$$

where $x_{0,b}$ and $x_{k,b}$ respectively are the DC and the kth AC coefficients of the bth DCT block,. $k \in S$, $S \subseteq \{1, 2,\ldots, 63\}$ is the set of index of DCT coefficients selected for watermark embedding by our position-selection algorithm, $x_{k,b-1}$ is the kth AC coefficient in the pervious block, α_s is watermark strength of the sth sub-image derived from SOM, Q_k is the quantization value for the kth AC coefficient in the default JPEG quantization table, and β is a scaling factor.

If a 512×512 image need to be watermarked, the code-length of the fingerprints is 24,576, and m is set to 64, the watermarking procedure is as the following: First, we divide the original image into 4096 8×8 blocks and apply DCT to them. Moreover, the original image is divided into 64 64×64 sub-images and fed into SOM to get 64 watermark strength α_s, $s \in \{1,\ldots, 64\}$. Fig. 7 gives one example of the obtained watermark strengths. The darker area of the boat has higher

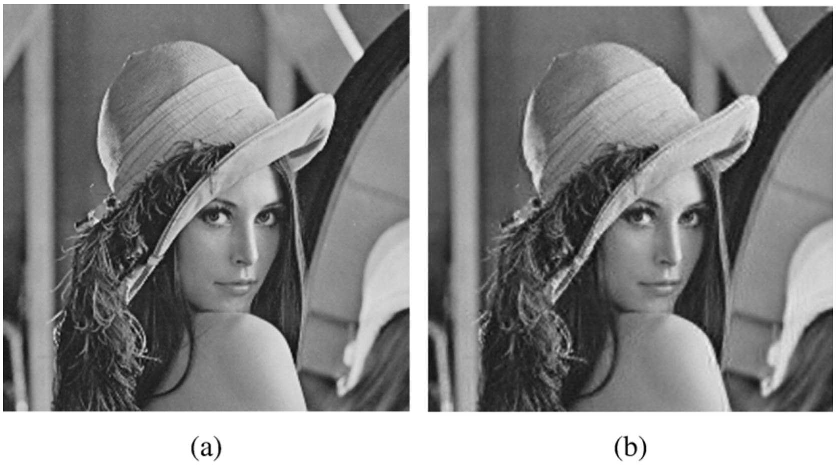


Fig. 13. Imperceptibility of the fingerprinting scheme: (a) The original image (256×256), and (b) One of the fingerprinted images.

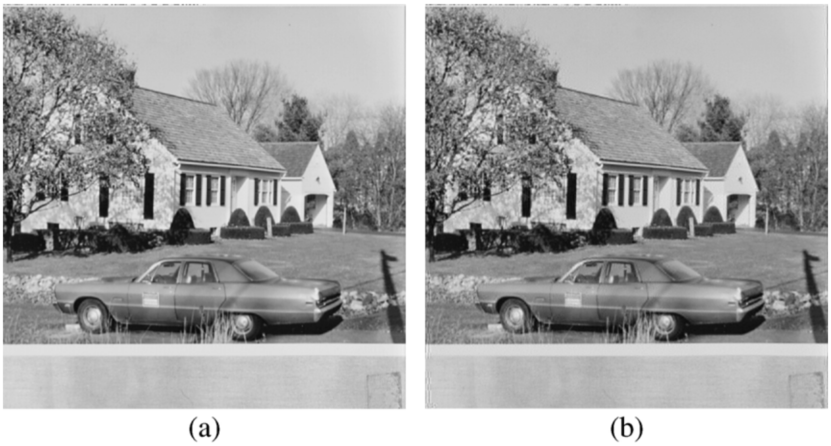


Fig. 14. Imperceptibility of the fingerprinting scheme: (a) The original image (512×512), and (b) One of the fingerprinted images.

tolerance to watermark addition and some sky and sea areas have lower tolerance.

The watermark can be extracted by simply comparing the values between $\widetilde{x}_{k,b}$ and $\widetilde{x}_{k,b-1}$. The watermark value is

$$\widetilde{v}_i = \begin{cases} 1, \text{if } \widetilde{x}_{k,b} > \widetilde{x}_{k,b-1} \\ 0, \text{if } \widetilde{x}_{k,b} < \widetilde{x}_{k,b-1} \\ ?, \text{otherwise} \end{cases}. \tag{6}$$

If the extracted value is “?”, we decode it as an erasure. The erasures should be tolerated by the traitor detection procedure, which will be introduced in the next section.

We measure the robustness of watermarking by the normalized correlation

$$\mathrm{NC} = \frac{\sum_i v_i \widetilde{v}_i}{\sum_i v_i^2}. \tag{7}$$

The relationship between robustness (*NC*) and imperceptibility (*PSNR*) is shown in Fig. 5. The size of test images is 256×256, and the fingerprint is a 16-bit code, and which is repeatedly embedded to 24 DCT coefficients, determined by the embedding position-selection algorithm presented in Section 4.3, for each of the 8×8 DCT blocks (i.e. 24576 DCT coefficients are modified in total). Without adaptive watermark-strength decision, the *NC-PSNR* point can be located at any points on the curve. Watermark strength should be assigned according to the trade-off between *NC* and *PSNR*, but it is impractical to decide the strength after analyzing the curve, furthermore, global assignments of watermark-strength on the whole image may ignore local characteristics of the image. The experiments show good results of our adaptive decision strategy since the three points resulting from different size of image-blocks all gather at the right-top corner of the curve, which means the decision is close to the optimal solution. In addition, the performance is the best when the block size *m* is 64, which seems reasonably because 128 is too coarse to localize the decision and 32 is too small to assign the strength precisely for humans while training. Fig. 8 illustrates the watermarks produced by adaptive strength-decision watermarking (ASDW), and Fig. 9 shows the difference between watermarks of fixed-strength watermarking (FSW) and ASDW. In Fig. 9(a), we find the difference between watermarks disperse in the whole image, while in Fig. 9(b) and Fig. 9(c), the difference is obviously larger in the high-activity or darker areas. This can be explained by observing the distribution of DCT coefficients. AC coefficients are larger in the high-activity blocks than in low-activity areas, therefore, the corresponding watermark-values are larger because they are obtained from the AC coefficients multiplied by the watermark-strength. In dark blocks, the DC coefficients are comparatively smaller, so slightly changes to AC values may cause an obvious effect in the spatial domain.

4.3. Selection algorithm for embedding positions

Collusion attack is the main menace to fingerprinting schemes. Besides the fingerprint-code designing, watermarking strategies are also important for collusion-resistance. In this section, we present the algorithm to select embedding positions (namely, the DCT coefficients for embedding fingerprints) in which the fingerprinting scheme will have good performance for collusion-resistance.

Table 2
Comparisons of robustness to JPEG compression (in terms of *NC* values) between ASDW and IA-DCT

Algorithm	Watermark length	Image	Quality factor *Q*					
			80	60	40	20	10	5
ASDW	24576	House	0.83	0.86	0.86	0.83	0.65	0.74
IA-DCT	24936	IM8	0.9	0.78	0.66	0.51	0.38	0.22

Table 3
Comparisons of robustness to JPEG compression+cropping (in terms of *NC* values) between ASDW and IA-DCT

Algorithm	Watermark length	Image	Quality factor *Q*				
			Crop	Crop+Q80	Crop+Q60	Crop+Q40	Crop+Q20
ASDW	24576	House	0.86	0.76	0.80	0.90	0.40
IA-DCT	24936	IM8	1	0.92	0.79	0.68	0.52

Table 4
Robustness to contrast enhancing and noise addition (in terms of *NC* values)

Algorithm	Watermark length	Image	Contrast enhancing (20%)	Contrast enhancing (40%)	Noise addition (5%)	Noise addition (20%)
ASDW	24576	House	0.85	0.81	0.66	0.53

There are many types of collusion attacks. One of the common attacks is the average attack, in which the colluders average their copies to obtain the new version. The AND-attack is similar to the average-attack. For example, if the 3-bit binary code for three users is {011,101,110} and colluders are user 1 and user 3, then the pirate code resulting from the AND-attack is 010. In fact, this code is an AND-resilient 3-secure code, which means after being attacked by l colluders ($l \leq 3$) using the AND operator, colluders can still be uniquely identified. But in most practical applications, colluders can simply create the pirate version by selecting one symbol for each position from all possible symbols in that position of all the codewords owned by colluders. For this case, the code mentioned above is not 3-secure anymore because colluders could not be uniquely identified after this "selection-attack". For instance, user 1 and user 3 can collude to produce the pirate code 111, while user 1 and user 2, or any other coalitions could also produce this code.

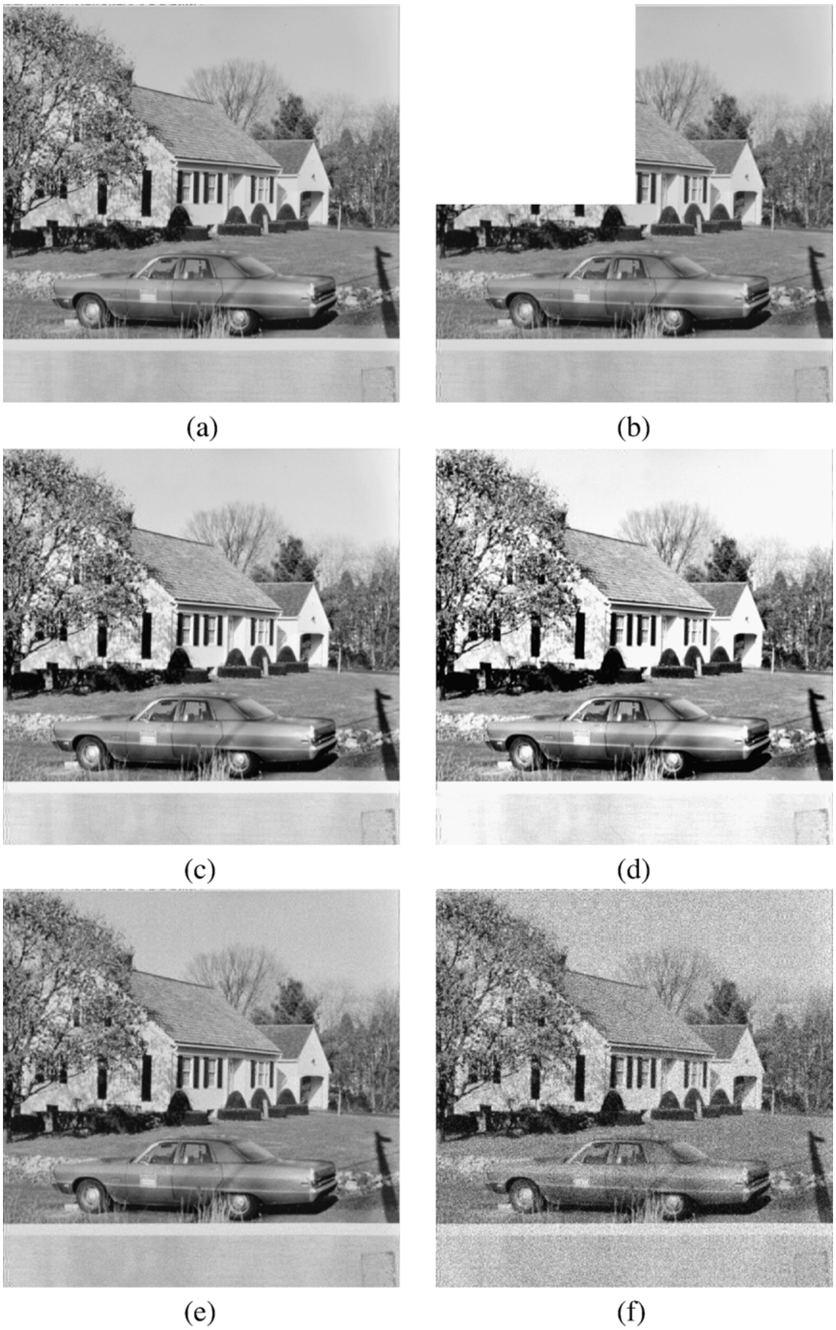


Fig. 15. The resulting watermarked images attacked by applying various image processing operations: (a) Original image (512 × 512), (b) Cropping (1/4), (c) Contrast enhancing (20%), (d) Contrast enhancing (40%), (e) Noise addition (5%), and (f) Noise addition (20%).

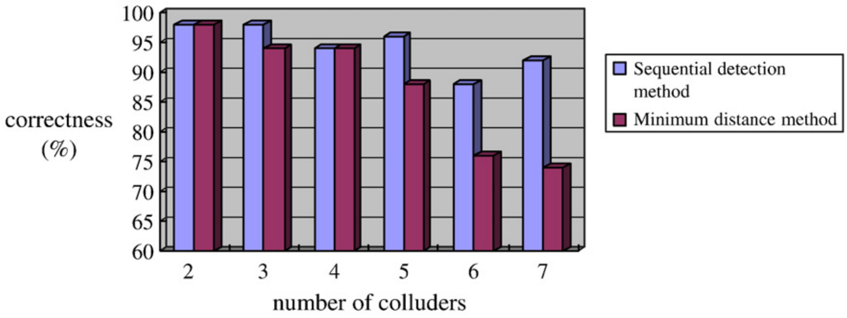


Fig. 16. The correctness of the traitor tracing if only one of the colluders must be identified.

Consequently, many existing fingerprinting codes will fail to resist the selection-attack. Fortunately, the ECC-based fingerprint codes do not have this drawback. The ECC-based fingerprinting scheme considers the attacks as noise and decodes the fingerprint based on the distance between the pirate code and the legal codewords. This distance-based decoding strategy can be applied to any types of collusion attacks. However, if all colluders contribute equal strength, then the pirate code has less traceable information because none of these colluders' codewords is obviously superior to others in the distance to the pirate code. One of the goals of fingerprinting scheme is to reduce the possible effect of this kind of information loss, that is, the "collusion-resistance", which can be achieved by designing the watermarking method properly. As mentioned in Section 1, randomness and cryptography are two common used techniques for collusion-resistance. Instead of applying these two techniques, we propose an optimization algorithm to achieve collusion-resistance. Our algorithm is based on the maximization of an energy function which indicates the collusion-resistance. The algorithm can be described as follows.

1. Choose applicable candidates, CP, for embedding positions, the embedding number n ($n \le |\mathrm{CP}|$), and the resilience-size c for fingerprint decoding and attacking simulations.
2. Initialize the embedding positions $EP^0=\{ep_{01}, ep_{02},\ldots, ep_{0n}\}$, $\forall i \in \{1,\ldots n\}$ $ep_{0i} \in CP$, and the energy function $E^0=f(R^0_{\mathrm{hit}}, R^0_{\mathrm{false}}, Q^0)$, where R^t_{hit} is hit ratio, R^t_{false} is false alarm rate, and Q^t is the imperceptibility of the watermark in the tth iteration (now the set of embedding positions is EP^t). R^t_{hit} and R^t_{false} are computed in a series of collusion-attacking simulations, while Q^t is estimated by watermarking experiments.

```
3. For $t=1,2,\ldots, t_{\max}$
    (1) Randomly re-select the set of embedding positions $\mathrm{EP}^t$ from CP.
    (2) Compute $R^t_{\mathrm{hit}}$, $R^t_{\mathrm{false}}$ and $Q^t$.
    (3) Calculate $E^t=f(R^t_{\mathrm{hit}}, R^t_{\mathrm{false}}, Q^t)$.
    (4) If $E^t < E^{t-1}$ then
    $R^t_{\mathrm{hit}} \leftarrow R^{t-1}_{\mathrm{hit}}$; $R^t_{\mathrm{false}} \leftarrow R^{t-1}_{\mathrm{false}}$; $Q^t \leftarrow Q^{t-1}$; $\mathrm{EP}^t \leftarrow \mathrm{EP}^{t-1}$;
    (5) If $|E^t - E^t - 1| \le \varepsilon$ then
    $T=t$; goto 5;
Next $t$.
4. $T=t_{\max}$;
5. Output $\mathrm{EP}^T$.
```

Our goal is to optimize the function value E^T, and we define energy function as $E^t=\alpha R^t_{\mathrm{hit}} - \beta R^t_{\mathrm{false}} + \gamma \mathrm{PSNR}^t$, that is, we use *PSNR* to be the estimation of watermark imperceptibility and to limit the distortion of image watermarking. In our experiments, CP is set as shown in Fig. 10(a), α is 0.2, β is 0.8, γ is 0.001, collusion size is 2, and 50 collusion attacks (in which colluder-

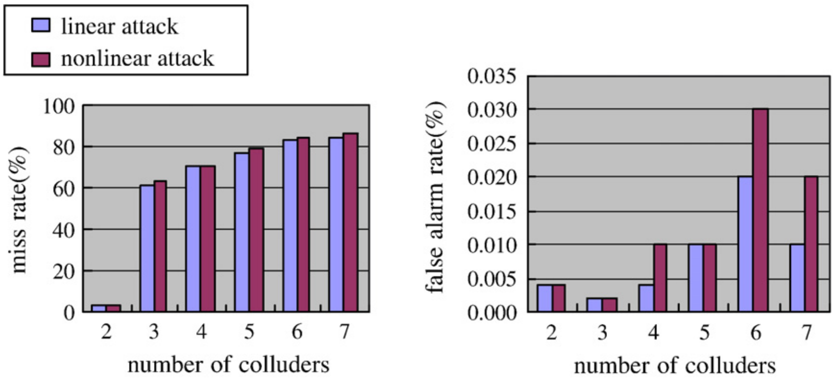


Fig. 17. Miss rates and false alarm rates of the colluder detections after linear and nonlinear collusion attacks, of collusion sizes from 2 to 7, if all possible colluders have to be identified.

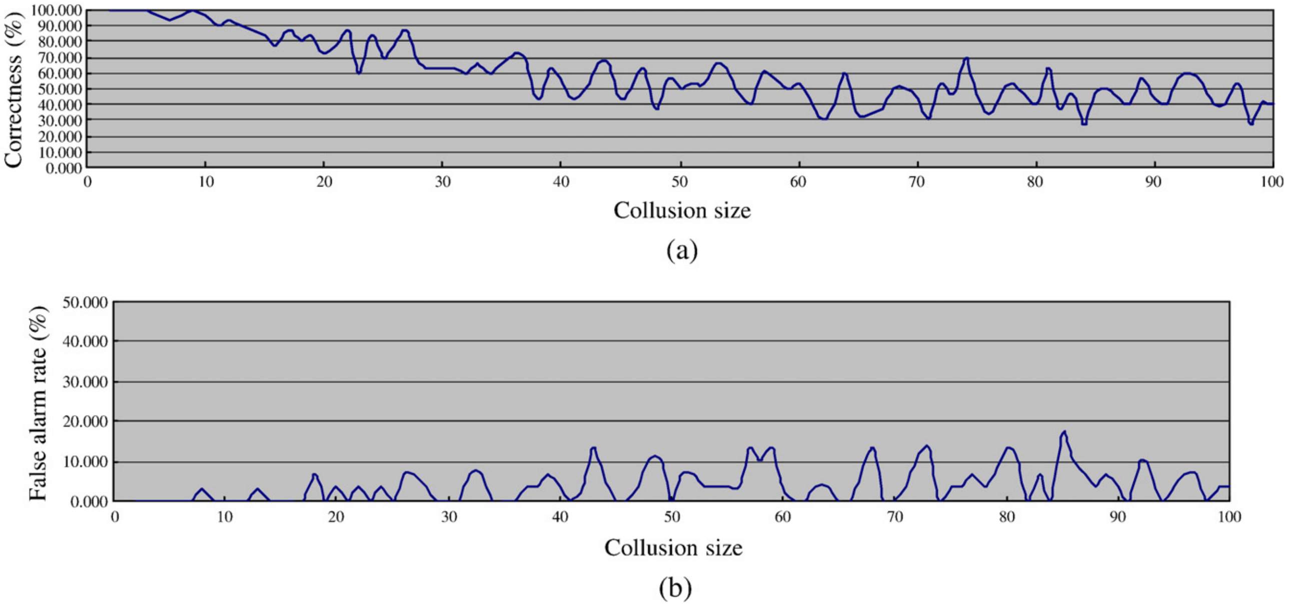


Fig. 18. The detection results of the traitor tracing for collusion sizes from 2 to 100: (a) The correctness, and (b) false alarm rates.

pairs are picked randomly) are performed to test the hit ratios and false alarm rates for traitor detection. Increasing of hit rates will also cause the raise of false alarm rates and the distortion of image quality, so the tradeoffs among $R_{\mathrm{hit}}{}^{t}$, $R_{\mathrm{false}}{}^{t}$, and Q^{t} must be considered when setting these parameters. One of the examples of EP^{t} is given in Fig. 10(b). After iterations of the proposed optimization procedure, the embedding positions are selected to maximize the energy function approximately. We test the performance of the algorithm for selecting embedding positions by computing the hit ratio and the false alarm rate. The test images is of size 256×256 and the fingerprint-code is still the same 24,576-bit code as used in previous sections. Fig. 11 illustrates the optimization process of our algorithm. Obviously, as the execution of iterations, the hit ratio increases, the false alarm rate decreases, and *PSNR* becomes larger progressively.

5. Sequential detection for traitor tracing

The marking assumption [1] in the fingerprinting problem says that users can only alter marks in detectable positions and should leave undetectable marks unchanged, but for multimedia, content values in undetectable positions are possibly disturbed by unintentional attacks (e.g., compression, geometrical alternations, noise addition). The marking assumption is then violated because the resulting content is affected by mixed attacks, even if the colluders do not intentionally modify the undetectable marks. As a result, the hard-decision decoding for ECC or other fingerprinting codes is inapplicable in reality. Moreover, *c*-TA codes only guarantee that at least one of colluders will be identified, but it would be more practical to trace the colluders as many as possible in most of the applications. We propose a sequential detection strategy tolerating erasures and errors to identify possible colluders by using the fuzzy membership rather than the crisp one.

5.1. Sequential detection algorithm

In order to narrow down the size of the candidates of colluders to reduce the overhead of the computation, we first select candidate-codewords within distance $L\text{-}L/c$ from the pirate code, in which L is the codeword length and c is the colluder size. Theorem 2 states that all codewords within this distance will be traitors. Since the watermarking process is lossy, the candidates are not certainly the colluders. We then define the set P as the positions where at least one of these candidates has mark "1", and the assurance matrix $A(b)(i)$, in which the assurance value indicate that user i is a member of colluders when the position indexed by the bth element in P is considered. $A(b)(i)$ is initialized as the value in the position indexed by the bth element in P for user i. For example, if the code is {1001,1011,0001,0111}, and the selected candidates are 1001 and 0001, then P is {0,3} and the initial $A(b)(i)$ is $\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$. That is, we only consider the $0th$ and the $3rd$ position of the codewords and use values in these positions for each user to estimate the assurance of belonging to colluders. Only candidates obtained in the previous stage are included for deriving P and A to find the users who own the codewords which are close to codewords of the candidates.

Theorem 2. [10] Suppose C is a code of length L, c is an integer, and the minimum distance d of C satisfies $D > L\text{-}L/c^{2}$. Then if C' is a coalition of size at most c, and w is the pirate code, then there exist an element of C' within distance L-L/c of w, and every codeword within distance L-L/c is in the coalition C'.

Our sequential detection strategy is presented as the following:

```
Procedure SequentialDetection
$C' = \varphi$
StillFound = True;
while StillFound do
    FindAtLeastOne = False
    while $t < |P|$ and not FindAtLeastOne do
        $m = m0 - t(m0 - 1)/|P|$;
        for $i = 1$ to $N$ do
            if $user(i) \notin C'$ then
                $\mu_t(i) = \delta_{t-1}(i)^{m+1}$;
                $A(t)(i) = (1 - \mu_t(i)) \cdot \delta_{t-1}(i) + \mu_t(i) \cdot 1$;
                $\delta_t(i) = \delta_{t-1} \cdot A(t)(i)$;
                if $\delta_t(i) > T1$ then
                    FindAtLeastOne = True;
                    break;
                end
            end
        end
        $t = t + 1$;
    end
    StillFound = False;
    for $i = 1$ to $N$ do
        if $\delta(i) > T2$ then add user(i) to $C'$
            StillFound = True;
        end
    end
```

Let N be the total number of users and C' is the set of colluders. The probability function of user i in the tth iteration is defined from the idea of fuzzy membership function as

$$\mu_t(i) = \delta_{t-1}(i)^{m+1} \tag{8}$$

in which $\delta_{t-1}(i)$ is the detection strength of user i in time $t-1$ for $i=1,\ldots,\ N$, and m is an integer used to increase the assurance of the membership iteratively. In the tth iteration, we consider the bit value in the position indexed by the tth element in P. If the current bit value is 1, it has assurance that the codeword is a little more close to the pirate code because this codeword agrees with the pirate code in the current position. The initial value of $\delta\ (i)$ is set to $\frac{M(i)}{L-e(i)}$, in which $e(i)$ is the number of erasures, and $M(i)$ is the number of matches between codeword i and the pirate word. The derived membership $\mu_t\ (i)$ is used to adjust the assurance matrix $A\ (t)(i)$, and then the new detection strength $\delta_t\ (i)$ is obtained. After considering all the bits indexed by P, users with $\delta_t\ (i)$ larger than a given threshold are decided to be traitors, next round of iterations continues until no users with high $\delta_t\ (i)$ are found.

The algorithm gives a run time complexity of $O(c|P|r)$, where $|P|$ denotes the cardinality of the set P and r is the decoding time of the fingerprint-code, which may be improved by using better decoding strategies of ECC.

6. Experimental results

256 × 256 and 512 × 512 images are used as the test data for our fingerprinting system. The fingerprints are constructed as described in Section 3. Without loss of generality, we consider only the sequential 2-TA code (i.e. $c=2$). The watermarking procedure is detailed in Section 4. That is to say, 24,576 coefficients in total of 256 × 256 or 512 × 512 DCT coefficients must be modified. This embedding ratio is much larger than 1000/256 × 256 in [25] and that of several other literatures [26,27], but our method still provides watermark transparency without losing robustness. Imperceptibility (Section 6.1) and robustness (Section 6.2) of watermarking methodology are examined through experiments, although our ultimate goal is to achieve high performance on traitor tracing. In experiments for traitor detection, collusion attacks are performed on fingerprinted images and the proposed sequential detection strategy is applied to trace as many colluders as possible. Although the theoretical resilience-size c is only 2, which is applicable in many applications, our method can still find at least one of the colluders with high correctness from pirate images produced by 3 or more colluders.

6.1. Imperceptibility

1000 codewords belonging to the first 1000 users are embedded into the host image to test the visibility of our system. If the image size is 512 × 512, the average *PSNR* of embedded images is 44.21 dB. But if the image size is 256 × 256, the average *PSNR* reduces to 32.49 dB because we have to embed much more bits in an 8 × 8 DCT block. Fig. 12 demonstrates the superiority of the adaptive watermark-strength decision in contrast to fixed-strength watermarking. As described in Section 4, watermark strengths will adapt to JNDs when ASDW is adopted. Fig. 13 shows the original image and one instance of the fingerprinted images (where image size is 256 × 256), and Fig. 14 gives another example for the case of image size 512 × 512. Watermarks in both of the two fingerprinted images are visually imperceptible.

6.2. Robustness

The size 512 × 512 House image (see Fig. 14(a)) is used to test the robustness. Tables 2, 3, and 4 show the *NC* values of extracted fingerprints after applying several different types of image processing operations to fingerprinted images and Fig. 15 shows the resulting images after attacks. For comparing our algorithm with the image-adaptive watermarking in the DCT domain (IA-DCT) with visual models proposed in [14], the same compression and cropping attacks are performed. However, IA-DCT is a pure "watermarking" scheme while our system is a "fingerprinting" scheme, as a result, the adjustments of parameters may have different purposes and the algorithms may have different considerations on the trade-offs between robustness and imperceptibility.

In Table 2, robustness to JPEG compression is examined. The erasure-bits are skipped when computing NC values. It is observed that ASDW does not get so good performance as that of IA-DCT for high quality factor Q, but for lower Q ($Q \leq 60$), ASDW performs better. This is because ASDW uses relative relationship between DCT coefficients in adjacent blocks to embed watermarks, rather than embeds an absolute value in the DCT coefficients; moreover, ASDW has taken quality factor into account in the embedding formula. Table 3 shows the results after cropping the original image followed by compression, which reports similar results as in Table 2. In Table 4, we find that ASDW can resist the contrast-enhancing well and has certain resistance to the noise-addition attack.

6.3. Correctness of traitor tracing

We randomly choose 50 coalitions for each of the 6 collusion sizes (from 2 to 7), then use the linear attack and the nonlinear attack to produce the pirate images. The linear attack used in this experiment is applying the average operation on all the images of colluders while the nonlinear attack taking the average of the minimum and the maximum values.

If only one of the colluders must be identified, we compare the sequential detection strategy with the method of capturing the user who owns the fingerprinted image with minimum hamming distance from the pirate image. From Fig. 16, the proposed sequential detection method is obviously superior to the minimum distance approach especially when the collusion size is large, in which “correctness” is defined as the probability of finding at least one of the colluders. If all possible colluders must be identified, the sequential detection algorithm is applied to all of the 600 tests. Fig. 17(a) shows that if collusion size is not smaller than c, then the miss ratio is acceptable, but when the collusion size is larger than c, the miss ratio raises severely. As illustrated in Fig. 17(b), all false alarm rates are near zero for all six sizes of collusion. This is due to the trade-off between miss ratios and false alarm rates. In a fingerprinting system, the detection correctness is the most important issue, so the high miss ratio when collusion size is larger than c does not imply heavy degradation on the performance. From Fig. 17(b), the false alarm rate for collusions of size 6 is worse than collusions of size 7, it is because the values of the membership for each user are near uniform for some test data of size 6, therefore, membership values for innocent users may be even slightly larger than that of the traitors due to some watermarking errors, as a result, false alarm rate increase abruptly. Fig. 18 gives the results of traitor tracing for collusion sizes from 2 to 100, which shows that if the collusion size is not much larger than c, the detection performance is still good. Comparing with the fingerprinting scheme based on orthogonal signals [28], which can be broken by only a few dozen of colluders and so the performance curve has steep descending slope when collusion size is about 30, the proposed fingerprinting system has much more smooth curve.

7. Conclusions

We have introduced a traceable content-adaptive fingerprinting system for multimedia. A c-TA code which can be constructed systematically and flexibly for multimedia is adopted with the cost of long code-length. Combining the proposed content-adaptive watermarking scheme with this code can achieve high *PSNR* of fingerprinted images while serving large number of users. The optimization procedure for selecting embedding-coefficients will pick suitable embedding positions to resist collusion attacks. The sequential detection method, which can tolerate erasures and errors, produces a high degree of correctness for traitor tracing. This fingerprinting scheme, with good performance, has been implemented with practical parameters, including applicable resilience-sizes and the size of customer base. It is our belief that the work done in this paper provides a feasible methodology for multimedia fingerprinting and traitor tracing.

Acknowledgement

This work was partially supported by the National Science Council and the Ministry of Education of ROC under the contract no. NSC92-2622-E-002-002, NSC92-2213-E-002-023, 89E-FA06-2-4-8 and NSC93-2752-E-002-006-PAE. We would also like to acknowledge Dr. Chun-Hsiang Huang for his helpful discussions and valuable suggestions.

References

[1] D. Boneh, J. Shaw, Collusion-secure fingerprinting for digital data, Proc. CRYPTO’95, 1995, pp. 452–465.

[2] B. Chor, A. Fiat, M. Naor, Tracing traitors CRYPTO’94, vol. 839, 1994, pp. 480–491 .

[3] A. Fiat, M. Naor, “Broadcast Encryption”, CRYPTO’93, pp. 480–491.

[4] D.R. Stinson, T.V. Trung, Some new results on key distribution patterns and broadcast encryption, Designs, Codes and Cryptography 14 (1998) 261–279.

[5] D.R. Stinson, On some methods for unconditionally secure key distribution and broadcast encryption, Design, Codes and Cryptography 12 (1997) 215–243.

[6] D.R. Stinson, R. Wei, Combinatorial properties and constructions of traceability schemes and frameproof codes, SIAM Journal on Discrete Mathematics 11 (1998) 41–53.

[7] J. Dittmann, P. Schmitt, E. Saar, J. Ueberberg, Combining digital watermarks and collusion secure fingerprints for digital images, JEI 9 (4) (2000) 456–467.

[8] W. Trappe, M. Wu, J. Wang, K.J.R. Liu, Anti-collusion fingerprinting for multimedia, IEEE Transactions on Signal Processing 51 (2003) 1069–1087.

[9] http://www.its.caltech.edu/~ssachan/math6b/design.pdf.

[10] A. Silverberg, J. Staddon, J.L. Walker, Applications of list decoding to tracing traitors, IEEE Transactions Information Theory 49 (2003) 1312–1318.

[11] H. Zhao, K.J.R. Liu, Bandwidth efficient fingerprint multicast for video streaming, IEEE International Conference on Acoustics, Speech and Signal Processing (2004).

[12] R. Safavi-Naini, Y. Wang, A code for sequential traitor tracing, ASPCS (2002) 211–224.

[13] R. Safavi-Naini, Y. Wang, Sequential traitor tracing, IEEE Transaction on Information Technology 49 (2003) 1319–1326.

[14] M.S. Kankanhalli, R.K.R. Ramakrishnan, Content based watermarking of images, ACM Multimedia’98, 1998, pp. 61–70.

[15] C.I. Podilchuk, W. Zeng, Image-adaptive watermarking using visual models, IEEE Journal on Selected Areas in Communication 16 (1998) 525–539.

[16] M.D. Swanson, B. Zhu, A.H. Tewfik, Multiresolution scene-based video watermarking using perceptual models, IEEE Journal on Selected Areas in Communication 16 (1998) 540–550.

[17] K.J. Davis, K. Najarian, Maximizing strength of digital watermarks using neural networks, IEEE International Conference in Neural Networks 4 (15–19) (July, 2001) 2893–2898.

[18] Z.M. Zilang, R.Y. Li, L. Wang, Adaptive watermark scheme with RBF neural networks, IEEE International Conference on Neural Networks & Signal Processing (Dec, 2003) 1517–1520.

[19] D. Kundur, K. Karthik, Video fingerprinting and encryption principles for digital rights management, Proceeding of the IEEE 92 (6) (June, 2004) 918–932.

[20] C.H. Huang, J.L. Wu, A watermark optimization technique based on genetic algorithms, SPIE Electronic Imaging 2000, San Jose, January, 2000.

[21] M. Fernandez, M. Soriano, Soft-decision tracing in fingerprinted multimedia content, IEEE MultiMedia 11 (2) (April/June 2004) 38–46.

[22] T. Kohonen, The self-organizing map, Proceeding of The IEEE 78 (9) (1990) 1464–1480.

[23] J.N. Staddon, D.R. Stinson, R. Wei, Combinatorial properties of frame-proof and traceability codes, IEEE Transaction on Information Theory 47 (3) (2001) 1042–1049.

[24] J.A. Freeman, D.M. Skapura, Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, 1992, p. 267.

[25] I.J. Cox, J. Kilian, F.T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Transaction on Image Processing 6 (1997) 1673–1687.

[26] C.T. Hsu, J.L. Wu, Hidden digital watermarks in images, IEEE Transaction on Image Processing 8 (1999) 58–68.

[27] Q. Cheng, T.S. Huang, An additive approach to transform-domain information hiding and optimum detection structure, IEEE Transaction on Multimedia 3 (2001) 273–284.

[28] Z.J. Wang, M. Wu, W. Trappe, K.J.R. Liu, Group-oriented fingerprinting for multimedia forensics, EURASIP, 2004, pp. 2153–2173.