2007

A maximum likelihood watermark decoding scheme

Wenming Lu
University of Wollongong, wl86@uow.edu.au

Wanqing Li
University of Wollongong, wanqing@uow.edu.au

Rei Safavi-Naini
University of Wollongong, rei@uow.edu.au

Philip Ogunbona
University of Wollongong, philipo@uow.edu.au

A maximum likelihood watermark decoding scheme

Abstract

Based on the observation that an attack applied on a watermarked image, from a decoding point of view, modifies the distribution of the detection values away from the ideal distribution (without attack) for corresponding watermarking scheme, we propose a generic maximum likelihood decoding scheme by approximating the distribution with a finite Gaussian mixture model. The parameters of the model are estimated using expectation-maximization algorithm. The scheme allows the decoding to be automatically adapted to attacks that the watermarked images have undergone and, in consequence, to improve the decoding accuracy. Experiments on a QIM based watermarking system have clearly verified the significant improvement of the decoding accuracy achieved by the proposed maximum likelihood decoding in comparison to conventional threshold decoding

Disciplines

Physical Sciences and Mathematics

Publication Details

This conference paper was originally published as Lu, W, Li, W, Safavi-Naini, R, Ogunbona, P, A Maximum Likelihood Watermark Decoding Scheme, 2007 IEEE International Conference on Multimedia and Expo ICME 2007, Beijing, 2-5 July 2007, 1247-1250. The original paper is available here.

A MAXIMUM LIKELIHOOD WATERMARK DECODING SCHEME

Wenming Lu, Wanqing Li, Rei Safavi-Naini, Philip Ogunbona

University of Wollongong, Australia

ABSTRACT

Based on the observation that an attack applied on a watermarked image, from a decoding point of view, modifies the distribution of the detection values away from the ideal distribution (without attack) for corresponding watermarking scheme, we propose a generic maximum likelihood decoding scheme by approximating the distribution with a finite Gaussian mixture model. The parameters of the model are estimated using expectation-maximization algorithm. The scheme allows the decoding to be automatically adapted to attacks that the watermarked images have undergone and, in consequence, to improve the decoding accuracy. Experiments on a QIM based watermarking system have clearly verified the significant improvement of the decoding accuracy achieved by the proposed maximum likelihood decoding in comparison to conventional threshold decoding.

1. INTRODUCTION

A typical image watermarking system consists of three major steps: 1) embedding a message into a host image; 2) the watermarked image undergoing an attack; 3) decoding the message from the attacked image. Research in the past has mainly focused on devising new or optimizing existing embedding schemes to achieve the desired embedding capacity and/or robustness against a set of attacks, such as additive noise, image processing and compression [1, 2, 3]. Characteristics of the attacks are sometimes taken into consideration in designing the embedding schemes. For instance, Local Average QIM (LAQIM) [3, 2] was designed to be robust against zero-mean additive noises and JPEG compression. However, decoding schemes are usually assumed to be simple and tightly bound to embedding schemes. Optimization of decoding against attacks has been virtually ignored [4]. This paper proposes a generic maximum likelihood (ML) approach, which automatically adapts to attacks through an unsupervised estimation of the way that the attacks modify the watermarked images, and is applicable to a variety of watermarking systems with little modifications.

Considering a scenario where a binary message, m_e, is to be embedded into an image. The image is first divided into blocks, and each block is then encoded with one bit of m_e using a chosen embedding algorithm. The watermarked image may undergo a number of attacks before it is communicated to the decoder where m_e is to be recovered. At the decoder, the detection algorithm associated with the embedding algorithm is employed to each block independently. It calculates a detection value from each block, compares it with a threshold and decides the embedded bit, 0 or 1, accordingly. The detection value may be a correlation coefficient or linear correlation in the spread spectrum system [1], or the distances to the closest bit 0 and 1 centroid in QIM. We refer this type of decoding as *threshold* decoding.

From decoding point of view, any attack to a watermarked image tends to change the detection values away from the ideal values and, therefore, modifies the distribution of the detection values. Ideal detection values are the detection values without any attack. Threshold decoding is to essentially compare the calculated detection values with the ideal values for bit 0 and 1. If an attack modifies the detection values in such a way that they form a symmetric distribution around the ideal detection value, threshold decoding shall continue to perform well to certain extent. However, attacks sometimes change the detection values in a non-symmetric manner. In this case, estimation of the statistical distribution of the detection values not only can improve the decoding, but may also reveal the types of attacks. Fig.1 shows the distributions of the detection values of all embedding blocks after the attack of JPEG70% or additive Gaussian $N(2,2)$, where the embedding scheme is LAQIM with $\Delta = 10$. The ideal detection values is concentrated at 0 for bit 0, -5 and $+5$ for bit 1. The curves show that JPEG tends to modify the detection values in a symmetric manner whereas $N(2,2)$ does not.

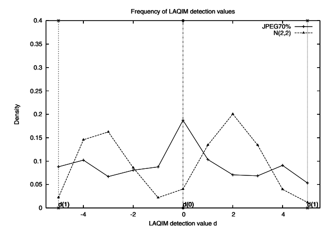

Fig. 1. Distributions of ideal detection values without attacks and detection values after JPEG 70% and Gaussian noise N(2,2) attacked $lena$. The embedding scheme is LAQIM with quantization step Δ=10

In this paper, we propose to model the distribution of the detection values with *finite gaussian mixture model* (FGMM) and to automatically estimate the related parameters using the *Expectation-Maximization* (EM) algorithm. A *maximum likelihood* (ML) decoding scheme is formulated using the estimated distribution. The FGMM does not assume any knowledge of the attacks that the watermarked image has undertaken and the EM algorithm tries to adapt FGMM to the unknown attacks.

The paper is organized as follows. Section 2 presents a generic ML decoding framework given the probability density function (pdf) of the detection values. Section 3 describes the Gaussian mixture modeling of the distribution and the estimation of its parameters using EM. In Section 4, experimental results of applying the proposed ML on LAQIM are presented. Section 5 concludes the paper with remarks and future work.

2. ML DECODING

Without losing generality, we consider embedding an n-bit message m_e in the spatial domain of an image I. I is partitioned into n em-

1-4244-1017-7/07/$25.00 ©2007 IEEE

bedding blocks. A watermarking algorithm, Γ, is chosen to embed m_e into I, one bit per block. In decoding, the corresponding detection algorithm, Γ^{-1}, calculates a detection value from each block of the watermarked image that may have been subjected to attacks. The detection values from all blocks form a *d-map*. Let D be a random variable and each detection value d is considered as an observation of the random variable D. The random process of D is governed by the attacks that the watermarked image has undergone. Let $p(D)$ be the probability density function (pdf) of D. If the respective proportions of the bit 0 and bit 1 in the message m_e are α_0 and α_1, where $\alpha_0 + \alpha_1 = 1$, then $p(D)$ can be written as

$$p(D) = \alpha_0 p_0(D) + \alpha_1 p_1(D), \tag{1}$$

where $p_0(D)$ and $p_1(D)$ are the respective pdfs of the detection values produced from blocks that are actually embedded bit 0 and 1.

Given a detection value, d, ML estimates a bit $b \in \{0, 1\}$ such that

$$b = \arg\max_{b\in\{0,1\}} \alpha_b p_b(d), \tag{2}$$

where $\alpha_b p_b(d)$ is the likelihood of the originally embedded bit to be bit b given the detection value d.

Obviously, the proposed ML decoding relies on the estimation of $\alpha_b p_b(d), b = 0, 1$. In the following section, we propose to model $p(D)$ using finite Gaussian mixture and estimate the underlying parameters using EM. The $\alpha_b p_b(d), b = 0, 1$ is obtained by heuristically partitioning the Gaussian densities that fit to $p(D)$ into two groups, one for $p_0(D)$ and another for $p_1(D)$, respectively.

3. FGMM AND PARAMETER ESTIMATION OF $P(D)$

We assume that $p(D)$ can be approximated by a finite Gaussian mixture model (FGMM) with k components [5], i.e.

$$p(D|\Psi) = \sum_{i=1}^{k} a_i f_i(D|\mu_i, \sigma_i), \tag{3}$$

where a_i is the mixture weight for the $i'th$ component, $\sum_{i=1}^{k} a_i = 1$; $f_i(\cdot)$ is a Gaussian function with mean μ_i and variance σ_i. Ψ denotes the parameter set $\{a_i, \mu_i, \sigma_i, i = 1, 2, \ldots, k\}$.

Now the problem becomes how to estimate the parameter set Ψ such that $p(D|\Psi)$ best fits the given *d-map*, $\mathtt{d}$, that consists of n detection values. Let $L(\Psi)$ be the total log-likelihood [5] of the n detection values fitting to $p(D|\Psi)$, i.e.

$$L(\Psi) = \mathtt{ln}\prod_{i=1}^{n} p(d_i|\Psi) = \mathtt{ln}P(\mathtt{d}|\Psi), \tag{4}$$

where $\prod_{i=1}^{n} p(d_i|\Psi) = P(\mathtt{d}|\Psi)$. Given a Ψ, $L(\Psi)$ measures the goodness of fit of $p(D|\Psi)$ to the observed *d-map*. Hence, maximization of $L(\Psi)$ with respect to Ψ, for a given *d-map* $\mathtt{d}$, yields the maximum likelihood estimation (MLE) of Ψ, i.e., the best fitted $p(D|\Psi)$. The problem of estimating $p(D)$ is then to produce a parameter set $\Psi\{a_i, \mu_i, \sigma_i, i = 1, 2, \ldots, k\}$ that maximizes $L(\Psi)$.

EM is an iterative algorithm that is popularly adopted to maximize the likelihood function $L(\Psi)$ [5]. Assume at the q iteration, there is a parameter set Ψ^q. The objective is to find a new parameter set Ψ that satisfies $L(\Psi) > L(\Psi^q)$. This goal is equivalent to maximizing the difference between

$$L(\Psi) - L(\Psi^q) = \ln P(\mathtt{d}|\Psi) - \ln P(\mathtt{d}|\Psi^q). \tag{5}$$

A hidden variable is introduced purely as an artifice for making MLE of Ψ tractable with the assumed knowledge of the hidden variable. Denote the hidden random vector as $\mathbf{Z}$ and a given realization as $\mathbf{z}$, the updated value Ψ_{q+1} can formally be updated as

$$\Psi^{q+1} = \arg\max_\Psi\{E_{\mathbf{z}|\mathtt{d},\Psi^q}\{\ln P(\mathtt{d}, \mathbf{z}|\Psi)\}\},$$

and clearly, the EM algorithm consists of two iterating steps:

(i) E-step: determining the expectation $E_{\mathbf{z}|\mathtt{d},\Psi^q}\{\ln P(\mathtt{d}, \mathbf{z}|\Psi)\}$;

(ii) M-step: maximizing the expression with respect to Ψ.

For Gaussian function $f_i(\cdot)$, the concrete updating rules of the weights, the mean values and the variances are,

$$\begin{aligned} a_i^{q+1} &= \frac{1}{n}\sum_{j=1}^{n} f(i|d_j, \Psi^q) \\ \mu_i^{q+1} &= \frac{\sum_{j=1}^{n} d_j f(i|d_j, \Psi^q)}{\sum_{j=1}^{n} f(i|d_j, \Psi^q)} \\ \sigma_i^{q+1} &= \frac{\sum_{j=1}^{n}(d_j - \mu_i^{q+1})^2 f(i|d_j, \Psi^q)}{\sum_{j=1}^{n} f(i|d_j, \Psi^q)} \end{aligned} \tag{6}$$

in which $f(i|d_j, \Psi^q) = \frac{f_i(d_j, \Psi^q)}{\sum_{l=1}^{k} f_l(d_j, \Psi^q)}$. The updating rules actually include both E-step and M-step in each iteration and the algorithm keeps iterating until convergence, i.e., $L(\Psi)$ reaching the maxima.

3.1. Determining the number of components k

The iteration equations in (6) give the best estimation of the parameter set Ψ of FGMM with k components. However, the fitness of $p(D)$ to the given *d-map* also depends on the number of components k in FGMM. Determining the best k for the given *d-map* is a classical problem with many existing solutions including likelihood ratio (LR) test, Akikie's Information Criterion (AIC) and Minimum Description Length (MDL). A good review and comparative study of this problem can be found in [6]. In this paper, we adopt the LR approach. Let $L_k(\Psi)$ be the total log-likelihood of fitting k component FGMM into the *d-map* using EM. We choose k such that

$$k = \arg\max_{k_s \le k \le k_e} L_k(\Psi) \tag{7}$$

where $[k_s, k_e]$ is a range of possible values of k that is sufficient to capture the characteristic of various attacks. Our experiments have shown that k usually ranges from 2 to 8 components for attacks.

3.2. Determining $\alpha_b p_b(D)$

Given an estimated k-component $p(D|\Psi)$ that best fits the *d-map*, we need to separate k components into two groups: one represents the distribution of the detection values with originally embedded bit 0 and the other represents the distribution of the detection values with embedded bit 1, i.e., determining $\alpha_b p_b(D), b = 0, 1$ for the ML decoding.

The fact that the ideal detection values for bit 0 and 1 are often distinguished themselves well from each other leads to a number of heuristical methods for determining the $\alpha_b p_b(D), b = 0, 1$. Let λ_0 and λ_1 be the ideal detection values for bit 0 and 1, respectively. The simplest method is to sort k components in an ascending order by mean values and group the Gaussians into two groups with equal number of Gaussians. It is easy to determine which group should

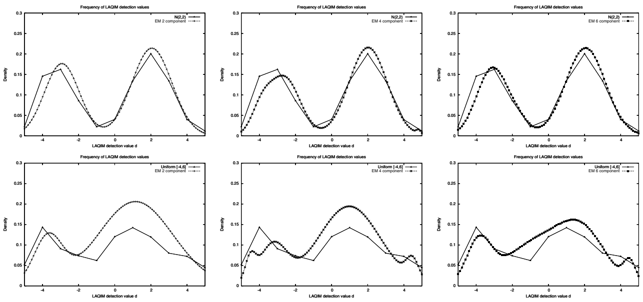

Fig. 2. Deciding k: the actual distribution is compared against assuming k=2, 4 and 6 from $left$ to $right$ against N(2,2) (Top) and Uniform[-4,6] ($Bottom$); in term of MLE, k=4 is the best fit and DER is also at the best 1.6% for N(2,2); also for Uniform[-4,6], k=4 is the best fit and DER is also optimal of 10.5%

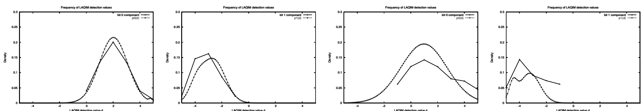

Fig. 3. estimated $p_0(d)$ and $p_1(d)$ after the partition against the actual frequency distribution of bit 0 and 1 components under N(2,2) ($left$) 2 figs, and under Uniform[-4,6] ($right$) 2 figs; for both estimation, k=4

be $p_b(D), b = 0, 1$ based on the relationship between λ_0 and λ_1. If $\lambda_0 \leq \lambda_1$, then the group with lower means belongs to $p_0(D)$. This method may not work well when the attack tends to influence the detection values from blocks encoded with bit 0 or 1 in a different way, or the bits in m_e is heavily biased to 0 or 1.

The second method is to cluster the k Gaussians into two groups. The Gaussians with means closer to λ_0 are said to belong to $p_0(D)$ and those with means closer to λ_1 form $p_1(D)$. That is

$$\begin{cases} |\mu_i - \lambda_0| < |\mu_i - \lambda_1| : f_i(\cdot) \Longrightarrow p_0(d) \\ |\mu_i - \lambda_0| \geq |\mu_i - \lambda_1| : f_i(\cdot) \Longrightarrow p_1(d). \end{cases} \quad (8)$$

Then α_0 and α_1 can be estimated as,

$$\begin{cases} \alpha_0 = \sum_u a_u, \forall u, f_u(d) \in p_0(d) \\ \alpha_1 = \sum_v a_v, \forall v, f_v(d) \in p_1(d). \end{cases} \quad (9)$$

where $u+v = k$. As in the first method, this simple clustering-based method assumes that the attacks modify the detection values moderately such that the relationship, between the detection values for both bit 0 and 1 and the ideal detection values, remains unchanged after attacks. If the attacks are strong enough to reverse the relationship, then the estimated $p_0(D)$ is actually for bit 1 and $p_1(D)$ for bit 0. The decoding bits will then be flipped.

To avoid bit flipping in the presence of strong attacks, additional information may be needed so that $p_b(D), b = 0, 1$ can be properly obtained from the estimated $p(D)$. Assume the proportions of the bit 0 and bit 1 in m_e are known, then the grouping of the Gaussians has to meet the constraint that the estimated α_0 matches the proportion of bit 0 contained in the message and the estimated α_1 matches that of bit 1 of the message. In this case, a search method of estimating parameters may be developed. It is feasible to employ Brute-Force search since k is usually small.

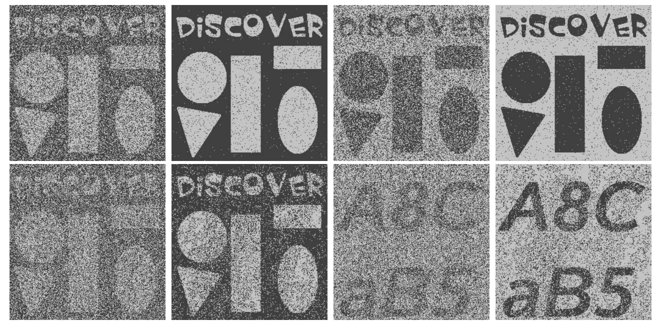

Fig. 4. (Top) under Gaussian from $left$ to $right$: N(2,2), TD-31.76%, ML-1.6%; N(2,3), TD-68%, ML-98% (bits are flipped under the attack; however, the quality of the logo is tremendously enhanced), TD-threshold decoding; ($Bottom$) two logos decoded from $lena$ under joint attacks by JPEG70% and N(2,2) from $left$ to $right$, TD-38.77% and ML-19.25%; TD-38.67% and ML-20.95%

4. EXPERIMENTAL RESULTS

In the section, we present results of applying the proposed ML decoding scheme to recover the binary logos that are embedded into grey-scale images using LAQIM [2]. We not only compare the ML decoding with conventional threshold decoding in terms of decoding accuracy, but also demonstrate how well the estimated $p(D)$ fits the real distribution of the detection values.

4.1. The LAQIM system and attacks

In LAQIM, an image is divided into n square blocks of size $B \times B$ to embed an n bit message m_e. Each bit is embedded into the average pixel intensity of a block using QIM [3]. In decoding, the distances between the average pixel intensity of the attacked block and the closest bit 0 and 1 centroid, d^0 and d^1, are calculated respectively. Conventional *minimum distance* decoding is,

$$\begin{cases} d^0 \geq d^1 \Longrightarrow \; bit\ 1 \\ d^0 < d^1 \Longrightarrow \; bit\ 0. \end{cases} \quad (10)$$

Since LAQIM is a one-dimension QIM, d^0 and d^1 are correlated, d^0+d^1=$\Delta/2$, where Δ is the quantization step used for embedding. We define the detection value d as, $d = d^0$, if the average pixel intensity is bigger than (right to) the closest bit 0 centroid, and $d = -d^0$ otherwise. The following threshold decoding is then equivalent to the minimum distance decoding,

$$\begin{cases} d \in [-\Delta/2, -\Delta/4), (\Delta/4, \Delta/2] \Longrightarrow \; bit\ 1 \\ d \in [-\Delta/4, \Delta/4] \Longrightarrow \; bit\ 0. \end{cases} \quad (11)$$

Additive Gaussian noises, $N(\mu, \sigma)$ with the mean μ and standard deviation σ, and uniform noises, $[l, u]$, where l and u represent the range of the noise level, are used to simulate attacks.

4.2. Experimental parameters

In our experiments, we set $B = 2$ and $\Delta = 10$. The PSNRs between the original and the watermarked images are around 38.8db. The chosen attacks include Gaussian noises: N(1,2), N(2,2) and N(3,2), and uniform noises: [-4,6], [-3,7], [-2,8] and [-1,9].

The decoding error rate (DER) is used to quantitatively compare the decoding performance, DER=$\frac{l_{err}}{n}$, where l_{err} is the number of the error bits in the decoded message. Noticed that when $DER > 50\%$, we could consider that the decoded message is bit-wise flipped and the original message can be recovered by simply XOR each bit with 1. In the case that the embedded message is a binary logo, the effective decoding error rate is $1 - DER$. In other words, the worst decoding performance is $DER = 50\%$ as bits are completely randomized.

Thirty 512×512 8-bit gray-scale images that represent a wide range of different types of images are carefully selected as the test images. Five 256×256 binary logos are selected as the embedding messages. It has to be pointed out that the proposed ML decoding also works for random messages. We use binary logos for the sake of easy visual justification of the decoding results.

In threshold decoding, the threshold values are set at $-\Delta/4$=-2.5 and $\Delta/4$=2.5. The bit 0 is decoded for the detection value within [-2.5,2.5] or bit 1 otherwise.

In the estimation of $p(D)$, the possible values for k are set to be 2, 4, 6 and 8. For each k, the EM algorithm is employed to find the best fitted parameters, Ψ. The EM algorithm is initialized with equal weights. All mean values and variances are randomly initialized to the values around the mean and variance of the given *d-map*, but guaranteed to be different with random perturbation. The EM estimation stops when the increment of $L(\Psi)$ in two consecutive iterations is less than 0.001. The $p(D)$ that gives the maximum total log-likelihood, $L(\Psi)$, is chosen as the final approximation.

4.3. Results

The experiments on $lena$, $pepper$, $f16$, $mandrill$ and $boat$ using all attacks demonstrate that DER is minimum when $L_k(\Psi)$ reaches the maxima. This verifies that our proposed method for estimating $p(D)$ and $p_b(D)$ works effectively. For example, $L(\Psi)$ are -139150, -138554, -139231 and -139308 against N(2,2) on $lena$ for k being 2, 4,6 and 8, respectively; the corresponding DERs are 2.2%, 1.6%, 4.8% and 4.8%. Clearly, $L(\Psi)$ is maximized when k=4 and also DER reaches the best of 1.6%. Under the attack of uniform[-4,6], $L(\Psi)$ are -159150, -151122, -1521391 and -152541 for k being 2, 4, 6 and 8, separately; the DERs are 18.8%, 10.5%, 14.7% and 14.7%, respectively. DER is again optimal at k=4 by which $L(\Psi)$ is also maximized. Fig.2 illustrates the estimated distributions with $k = 2, 4, 6$ and the actual distribution of d after Gaussian noise attack N(2,2) and uniform[-4,6]. Fig.3 shows the estimated $p_0(d)$ and $p_1(d)$ using the clustering-based partition versus the ground truth.

Table 1 and 2 highlight the decoding results averaged on 30 images against the additive Gaussian and uniform noise attacks. The results show that ML outperforms threshold decoding significantly. Fig.2 shows a few decoded logos from the host image $lena$. Fig.3 also demonstrates two logos under joint attack of JPEG 70% and Gaussian noise $N(2, 2)$ in which ML decoding provides substantially better decoding than the threshold decoding.

Note that under attacks of N(3,2) and uniform[-2,8], DER is close to 100% and most bits in the decoded logo are flipped. This is due to the clustering-based method allocating the components to the opposite pdfs, $p_0(d)$ and $p_1(d)$. However, the visual quality of the logo remains greatly improved. To avoid bit flipping, the partition method of determining $p_b(D), b = 0, 1$, needs further optimization with or without the prior knowledge on α_0 and α_1.

$Gaussian\ attack$	$Threshold$	ML
$N(1,2)$	7.6%	3.1%
$N(2,2)$	31.8%	1.6%
$N(3,2)$	68.5%(31.5%)	97.4%(2.6%)

Table 1. DER under Gaussian; effective DER in bracket as flipping happens

$Uniform\ attack$	$Threshold$	ML
$[-4,6]$	16.8%	10.5%
$[-3,7]$	37.4%	9.9%
$[-2,8]$	63%(37%)	89.1%(10.9%)
$[-1,9]$	83.3%(16.7%)	91%(9%)

Table 2. DER under uniform; effective DER in bracket as flipping happens

5. CONCLUSION

In this paper, we proposed a generic ML image watermarking decoding scheme by approximating the distribution of detection values with a finite Gaussian mixture model and estimating the parameters of the model using the EM algorithm. The scheme is able to adapt itself to attacks and produce accurate decoding. Experimental results on LAQIM have clearly demonstrated the significant improvement of the decoding accuracy compared to conventional threshold decoding. More experiments on image processing-based attacks are being conducted and will be reported in the near future.

It is obvious that the grouping of Gaussians into $p_b(D), b = 0, 1$ is an important step in our proposed ML decoding method. The clustering approach is usually enough for moderate attacks and small k. Also in the paper, the components are fixed at Gaussian. The research of using large k for complicated attacks, and other distribution functions as the components, will be explored and addressed in the future work.

6. REFERENCES

[1] I.J.Cox, J.Killian, T.Leighton, and T.Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[2] W. Lu, W. Li, R. Safavi-Naini, and P. Ogunbona, "A new qim-based image watermarking method and system," *2005 Asia-Pacific Workshop on Visual Information Processing, Hong Kong*, pp. 160–164, 2005.

[3] B. Chen and G.W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Information Theory*, vol. 47, pp. 1423–1433, May 2001.

[4] M.Xia and B.Liu, "Effect of jpeg compression on image watermark detection," pp. 1981–1984, ICASSP, 2001.

[5] D.M.Titterington, A.Smith, and U.Makov, "Statistical analysis of finite mixture distributions," *John Wileys & Sons*, 1985.

[6] J.C.Bezdek, W. Q.Li, Y.Attikiouzel, and M.Windham, "A geometric approach to cluster validity for normal mi tures," *Soft Computing*, vol. 1-7, pp. 166–179, Sept. 1997.