Watermark detection based on the properties of error control codes

P. Loo and N. Kingsbury

Abstract: Watermark detection is a topic which is seldom addressed in the watermarking literature. Most authors concentrate on developing novel watermarking algorithms. In a practical watermarking system, however, one must be able to distinguish between watermarked and unwatermarked documents. Many existing systems belong to the class of so called ‘yes/no’ watermarks, where the detector correlates the candidate image with some known sequence to determine whether a mark is present. Unfortunately, these watermarks often carry no extra information and are not very useful. On the other hand, multi-bit watermarking schemes typically use a separate reference watermark and the payload of the watermark is decoded only when this reference watermark is successfully detected in the received image. It is shown that it is not necessary to use a reference watermark for detection purposes if the watermark payload is encoded with an error control code. One can then put all the energy into the payload watermark and increase its robustness. The turbo code is used as an example of error control codes in the work presented, and simulation results using an algorithm based on the authors’ previous work verifies their theory.

1 Introduction

The aim of watermark detection is to determine the likelihood of the presence of a watermark in an image, regardless of the content of the mark. Most existing robust watermarking systems fall into two categories: the so called 1-bit or ‘yes/no’ watermarks and multi-bit watermarks. In the former case, a single sequence is embedded into an image as the watermark and the watermark detector correlates the received image with this known sequence to determine whether a watermark is present [1, 2]. However, these watermarks often carry no extra information [Note 1] and are thus not very useful. In the latter case, although the watermark carries information, the decoder should only decode the payload if the watermark is successfully detected in the candidate image. The problem of watermark detection can be formulated into a hypothesis test:

\[ H_0: y = e \quad \text{watermark is absent} \]
\[ H_1: y = \eta s + e \quad 0 < \eta \leq 1 \quad \text{watermark is present} \quad (1) \]

where \( e \) is the noise vector, which may include the host image coefficients, and \( s \) is our watermark signal. \( \eta \) takes into account the possibility that the watermarked image is attacked and the amplitude of the watermark \( s \) is reduced as a result. Lu et al. [3] proposed the use of the locally optimal detector (LOD) which maximises the slope of the power function when \( \eta \) is close to zero. If the noise statistics are known or can be estimated, the corresponding LOD can be derived as

\[ -\sum \lambda \frac{P_f(y)}{P_d(y)} \frac{h_1}{h_0} \geq \lambda \quad (2) \]

where \( P_f(y) \) and \( P_d(y) \) are the probability density function (pdf) of the noise statistics and its derivative. The threshold \( \lambda \) can be chosen according to the Neyman–Pearson principle [4] by fixing the false positive probability \( P_{FP} \). Hernández et al. [5, 6] analysed watermark detection using correlation in a spread spectrum system under filtering attack, and derived a Neyman–Pearson based detector. The authors proposed an approach where the watermark is detected in ‘one shot’ without first being decoded. In this paper, we will assume the watermarked image has not been attacked in order to simplify our analysis. In the case of watermark decoding, usually no threshold is necessary, as decoder statistics are often symmetrical about zero, especially if the decoder is correlation based, and so we can conveniently use zero as the threshold in distinguishing between bit 1 and bit 0.

Many existing watermark detectors assume a portion of the watermark is known in advance. This is satisfied in yes/no watermarks, where the detector knows exactly what the embedded sequence is. If we have a multi-bit payload, watermark detection can be achieved by either of the following:

(i) Fix \( L \) bits of the payload to be of known value. The watermark decoder decodes the watermark as usual and compares these \( L \) bits to the known pattern. If all \( L \) bits match, then the watermark is detected.

Note 1: If there is only one watermark, then there is no extra information. However, we can have a set of watermarks and the detector detects each one in turn and returns the most probable one, if the detector output exceeds the threshold. The identity of the detected watermark carries information. Unfortunately, as the size of the set increases, this form of detection becomes impractical.
(ii) Allocate part of the payload energy to a separate watermark, which is embedded as a reference sequence. A watermark detector (e.g. LOD) is used to detect this reference watermark and the decoder proceeds only when the detector successfully detects the reference mark.

The first approach is used in multi-bit schemes such as [7]. The second approach is equivalent to the insertion of a template into a watermarked image, which is used in (e.g. [8, 9]) to combat geometric attacks. The template is actually a form of reference watermark and can be used for detection [Note 2]. In Sections 2 and 8 it is shown that these two approaches are approximately equivalent. We also derive the false positive \( P_{FP} \) and detection \( P_{D} \) probabilities in each case.

2 Watermark detection using reference watermark

As discussed in the introduction, we can detect a watermark by fixing part of the payload (say \( L \) bits) to some known pattern, or embedding a separate known sequence in addition to the actual watermark. The false positive and detection probabilities of the two cases are derived in the Appendix and are shown as follows. We assume no error control code (ECC) is used for the moment.

Case 1, \( L \)-bit fixed pattern [Note 3]:

\[
P_{FP} = 2^{-L}
\]

\[
P_D = \left[ 1 - Q \left( \sqrt{\frac{M}{L+M} \cdot \text{SNR}} \right) \right]^L
\]

where \( M \geq 1 \) is the length of the actual payload, \( \text{SNR} \) is the signal-to-noise ratio per bit \( (E_b/No) \) at the decoder output if no reference watermark is used and \( Q(x) \) is the tail integral of the Gaussian distribution defined as

\[
Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} \exp \left( -\frac{x^2}{2} \right) dx
\]

\( Q^{-1}(\cdot) \) is the inverse of (5). In the case of a yes/no watermark, the first approach is equivalent to splitting the watermark into \( L \) bits and using a watermark decoder for detection. The false alarm rates for both approaches stay the same and the detection probabilities become:

Case 1, \( L \)-bit fixed pattern:

\[
P_{D_{\text{yes/no}}} = \left[ 1 - Q \left( \sqrt{\frac{E}{L \cdot No}} \right) \right]^L
\]

Case 2, detecting entire sequence as a whole:

\[
P_{D_{\text{yes/no}}} \approx 1 - Q \left( \sqrt{\frac{E}{No} - Q^{-1}(p)} \right)
\]

Note 2: In [9], the authors proposed a method to detect both the presence of the watermark and the payload length using a Bayesian framework. Note 3: The false positive probability listed here is based on a hard decoder, see the appendix (Section 8.1).

Fig. 1 Receiver operation characteristics (ROC) of 128-bit watermark at ratio \( E_b/No \) of 2 dB when either fixed \( L \)-bit pattern or single reference watermark is used for watermark detection

where \( E_b/No \) is the ratio of the watermark energy to the power spectral density (PSD) of the noise (assumed to be white). Fig. 1 shows the receiver operation characteristics (ROC) curve for a 128-bit watermark at a ratio \( E_b/No \) of 2 dB. We can see that using a separate reference watermark is better than using a fixed \( L \)-bit pattern as expected [Note 4]. In spread spectrum systems, a reference watermark can simply be superimposed on the actual watermark, as long as the reference mark is orthogonal to the embedded watermark. However, the reference watermark consumes some of the watermark energy and reduces the robustness of the payload watermark. Fortunately, we can use error control codes for watermark detection, as will be demonstrated later.

3 Error control codes and their properties

Error control codes (ECC) are often used to encode the payload prior to watermark embedding, to improve the decoded error rate. A commonly used ECC in watermarking systems is the turbo code, which is a systematic recursive convolutional code using two or more coders concatenated in parallel. Turbo codes have been shown to achieve close to the Shannon capacity bound [10] and have found applications in areas where the signal-to-noise ratio is very low. The operation of turbo codes and their use in watermarking can be found in [11] and will not be discussed here.

Every ECC has a particular error correcting threshold [Note 5]. When the input error rate is below the threshold, the code can correct virtually all errors. On the other hand, when the error rate exceeds this threshold, the ECC decoder fails catastrophically and the decoded bitstream appears random. Depending on the decoding method and the nature of the error, the performance of a code degrades in different ways.

Fig. 2 shows the performance of two turbo codes under additive Gaussian white noise (AWGN) [Note 6].

Note 4: If error control codes are used for detection instead, then \( P_D \) is effectively 1 at \( E_b/No = 2 \) dB for the range of false positive probabilities shown, which can be deduced from Fig. 4.

Note 5: This threshold is related to the Hamming distance or \( d_{\text{free}} \) of the code, as well as the decoding algorithm, but discussion of this is beyond the scope of this paper.

Note 6: The result of the 16 384 bit code shown in Fig. 2 is taken from the JPL turbo code page http://www33.jpl.nasa.gov/public/JPLtcodes.html.
code parameters are two constraint length 5 encoders, interleaver block lengths of 128 and 16384 bits, respectively, and 10 decoding iterations. The input bit error rate (BER) of the 128-bit code, as well as the BER of an equivalent uncoded BPSK system are also shown.

The centre of each k-sphere is the corresponding codeword within the decoding region of a k-sphere which will be decoded to a k-bit payload corresponding to the centre of that k-sphere. This corresponds to the input BER being within the error correcting capability of that error control code (the radius of a k-sphere). An n-bit pattern extracted from an unwatermarked image will most probably fall into the gaps between the k-spheres. The ECC decoder will decode a k-bit pattern whose k-sphere is closest to the n-bit pattern. However, the distance between the received n-bit pattern and the centre of that k-sphere, which is the codeword after decoding the decoded k-bit pattern, will most probably be greater than the radius of the k-sphere, so we know the decoded payload is not reliable and the image is probably unwatermarked. The radius of the k-spheres is in effect our detection threshold. As we are using the payload itself for detection, we do not waste energy in embedding a reference watermark.

We shall now calculate the probabilities of false detection of a nonexistent watermark and of failure to detect an existing watermark. Given a threshold BER of \( P_e \), a false positive will occur if a random n-bit sequence has at least \((1 - P_e)n\) bits (assumed to be integer for the moment) in agreement with one of the codewords. The probability that exactly \( i \) of the \( n \) bits of a random sequence agree with some pattern is

\[
P(i \text{ out of } n) = \binom{n}{i} \frac{1}{2^n} \frac{1}{2^{n-i}} = \binom{n}{i} \frac{1}{2^n}
\]

where

\[
\binom{n}{i} = \frac{n!}{i!(n-i)!}
\]

is the number of possible ways of choosing \( i \) out of \( n \). The probability of false positive given a particular codeword is, therefore,

\[
P(\text{FP|some codeword}) = \sum_{i=(1-P_e)n}^{n} \binom{n}{i} \frac{1}{2^n}
\]
As the codeword closest to the extracted n-bit pattern is always chosen and the k-spheres do not overlap, the unconditional false positive probability is just (9) multiplied by the number of codewords:

\[ P_{FP,ECC} = \sum_{i=1}^{n} P(FP | \text{some codeword}) \]

\[ = \sum_{i=1}^{n} \binom{n}{i} \left( 1 - P_b \right)^{n-i} \]

where \( P_b \) is the probability that any given bit is decoded wrongly. Assuming all codewords are equally likely, the conditional and unconditional miss probability will be the same and are given by

\[ P_{Miss,ECC} = \sum_{i=1}^{n} \binom{n}{i} P_b^i (1 - P_b)^{n-i} \]

The probability of detection is, therefore,

\[ P_{Detect,ECC} = 1 - P_{Miss,ECC} \]

\[ = \sum_{i=0}^{n-1} \binom{n}{i} P_b^i (1 - P_b)^{n-i} \]

When the length of the codeword \( n \) is fixed, the false positive probability depends on \( k \) and \( P_e \), whereas the detection probability depends on the input bit error rate \( P_e \) and \( P_b \). The user first chooses the desired maximum allowable output BER of the decoder, the corresponding input BER gives an upper limit of \( P_e \). If we desire a maximum allowable output BER of say \( 10^{-4} \), this requires \( P_e \) to be smaller than 0.15 and the false positive probability will be at most \( 10^{-9} \). The detection performance will improve if we embed a longer watermark in a larger image, such that the energy per payload bit remains the same, because the gradient of the output BER increases with payload length (see Fig. 2).

As mentioned in the introduction, it is possible to detect a watermark in ‘one shot’ using a maximum likelihood (ML) detector. However, as the set of possible watermarks increases in size, the ML detector becomes rather complicated. Detection using ECC may not be optimal, but, as the ECC decoders use soft outputs between each iteration, the loss of performance compared with an optimal ML detector will be small. Besides, our approach makes it easier to select the threshold. In addition, as the detection stage is

---

Note 7: This is the code with block length of 128 bits in Fig. 2. As the constraint length of the encoder is 5, we need 4 bits of padding and the length of the payload is thus 128 - 4 = 124 bits.
independent of the watermark embedding/extraction process, this technique can be used to detect watermarks in any given watermarking system.

We perform simulations using a spread spectrum watermark system in the complex wavelets domain proposed in our previous work [13]. A 124-bit payload is encoded with the rate 1/3 turbo code to form a 384-bit coded watermark, which is then embedded into the ‘Lena’ image. The resulting PSNR is 38 dB. The watermarked image is JPEG compressed and the watermark is detected using the method described here. Figs. 6, 7 and 8 show the input BER to the turbo decoder, the output BER of the payload and the probability of detection of the watermark for a few values of $P_e$ under different JPEG quality factors, respectively. The output BER drops below $10^{-5}$ when the input BER is less than about 0.13, and so we can just set $P_e$ to 0.13 to maximise the detection probability. The transition from 0 to 1 in detection probability is very sharp around the chosen breaking off point of the turbo code, which justifies the use of error control codes in watermark detection. The threshold input BER under JPEG is slightly lower than that in Section 3 because the interference due to compression is not Gaussian. We did not simulate the false positive scenario because that would involve detecting watermarks from a large database of images, which is impractical.

5 Conclusions

In this work, we have addressed the problem of watermark detection, a topic which is often neglected in the literature. We have proposed an algorithm for watermark detection using error control codes. Our approach removes the need for a reference watermark for detection. The user chooses a maximum allowable output BER, and then selects the threshold input BER as a trade-off between false alarm and detection probabilities. Apart from detection, reference watermarks are also used for registration to combat geometric distortion attacks. However, if we construct our watermark sequences in a special way [14], the payload watermark can be made to be self-synchronising. As we no longer need a reference watermark, we can put all the available energy into the payload watermark to increase its robustness.

6 Acknowledgment

This work was supported by the Internal Graduate Studentship generously provided by Trinity College, Cambridge.

7 References


8 Appendix: Theoretical analysis of watermark detection using reference watermarks

We derive here the false positive and detection probabilities of the two approaches for watermark detection described in Section 2, namely fixing $L$ bits of the
payload to some known pattern or allocating part of the watermark energy to a separate reference watermark. We assume no error control coding is used here to encode the payload. There are two possible situations where a false positive can occur:

(i) The watermark detector declares a watermark is present when in fact there is none.
(ii) The watermark detector declares a watermark is present when in fact another watermark marked with the wrong key is present.

We can modify our null hypothesis to cope with the two cases as follows:

\[ H_0: \ y = e \quad \text{no watermark} \]
\[ H_0^*: \ y = s^* + e \quad \text{wrong key} \]  
(13)

where \( e \) and \( s \) are the noise (which includes the host signal) and the desired signal, respectively. From (1) is dropped for the sake of clarity. \( s^* \) is another watermark with the wrong key, such that \( E(s \cdot s^*) = 0 \).

The statistics of the correlator under these hypotheses are all Gaussian and have the following properties:

\[ \mu_{H_0} = \mu_{H_0^*} = 0; \quad \sigma_{H_0}^2 \approx \sigma_{H_0^*}^2; \quad \sigma_{H_0}^2 > \sigma_{H_0}^2 \]
(14)

\[ \sigma_{H_0}^2 \approx \sigma_{H_0}^2 \] because the image energy is much higher than the watermark energy and \( E(s^* \cdot e) = 0 \). As \( \sigma_{H_0} > \sigma_{H_0^*} \), it is the worst case scenario and we should use \( \sigma_{H_0^*} \) in our calculations. In the following analysis, we assume the energy of the payload is \( E_b \) per bit and that the noise power spectral density is \( N_0 \) (i.e., assuming white noise). We also assume the length of the payload is \( M \) bits. Therefore the SNR per bit is given by \( E_b/N_0 \) and the total energy \( E \) of the watermark is \( M \cdot E_b \).

### 8.1 L fixed bits

It can easily be seen that the decoder output will be symmetrical around the midpoint between bit 1 and 0 in an unwatermarked image (\( \mu_{H_0^*} = 0 \)). A simple but suboptimal way to detect a watermark is to use a bit-by-bit hard decoder, in which case the probability of getting \( L \) bits correctly at random (and hence false alarm) is simply,

\[ P_{FP} = 2^{-L} \]
(15)

If we require a false alarm rate of say \( 10^{-6} \), then we need \( L \geq 20 \). Alternatively, we can correlate the soft outputs of the decoder with our expected pattern and compare the results with a threshold (chosen, for example, via Neyman–Pearson), but this is equivalent to the case of using a separate reference watermark in the next section. In order to keep the distortion level in the watermarked image the same, the energy of the payload has to be reduced by a factor of \( M/(L + M) \). The effective energy per bit and the effective SNR per bit are

\[ E_b = \frac{M}{L + M} E_b, \]
\[ SNR' = \frac{M}{L + M} SNR \]
(16)

The new bit error rate can be derived using the new SNR (assuming the decoder output follows a Gaussian distribution) as

\[ P_{error} = Q\left( \frac{M}{L + M} \cdot SNR \right) \]
(17)

where \( SNR \) is the ratio \( E_b/N_0 \) at the correlator, in the absence of the fixed pattern, and \( Q(.) \) is defined in (5). Assuming the decoder decodes 1 bit at a time, we will miss a watermark if any of these \( L \) bits is decoded wrongly. Hence, the probability of detection is

\[ P_D = P(\text{all } L \text{ bits are decoded correctly}) = \left( 1 - Q\left( \frac{M}{L + M} \cdot SNR \right) \right)^L \]
(18)

### 8.2 Separate reference watermark

If we allocate some of our watermark energy to a separate reference watermark, say, by scaling the reference and the payload by \( a_1 \) and \( a_2 \), such that \( \sigma_1^2 + \sigma_2^2 = 1 \), so as to keep the resulting distortion the same. The payload total energy and the reference watermark energy are \( \sigma_1^2 E \) and \( \sigma_2^2 E \), respectively. If we equate the total energy of the reference watermark in the two cases, we can see that this is equivalent to setting \( L \) to

\[ L = \frac{\lambda M}{1 - \lambda} \]
(19)

As the entire reference watermark is detected as a whole, the effective ratio \( E_b'/N_0 \) of the reference watermark is

\[ E_b'/N_0 = \frac{\lambda}{1 - \lambda} \]

The user chooses the desired false alarm rate, say \( \rho \), and computes the threshold \( \lambda \) based on the Neyman–Pearson approach. The relationship between \( \rho \) and \( \lambda \) is:

\[ P_{FP} = \rho \]
(21)
\[ \lambda = \sigma_{H_0^*} Q^{-1}(\rho) \]
(22)

where \( Q^{-1}(\cdot) \) is the inverse of the \( Q \) function defined in (5). The watermark will be missed if the output of the correlator is less than \( \lambda \). The miss probability can be derived as follows:

\[ P_{Miss} = P(\text{output} < \lambda) \]
\[ = Q\left( \frac{\|H_0\|}{\sigma_{H_1}} - \lambda \right) \]
\[ = Q\left( \frac{\|H_0\|}{\sigma_{H_1}} - \frac{\sigma_{H_1} Q^{-1}(\rho)}{\sigma_{H_1}} \right) \]
\[ \approx Q\left( \frac{LM}{L + M} \cdot \frac{E_b}{N_0} - Q^{-1}(\rho) \right) \]
(23)

The last line follows from (14) and (20). The probability of detection is just

\[ P_D \approx 1 - Q\left( \frac{LM}{L + M} \cdot \frac{E_b}{N_0} - Q^{-1}(\rho) \right) \]
(24)
8.3 Special case: a yes/no watermark

In the case of a yes/no watermark (i.e. $M = 0$), the above expressions of detection/miss probabilities need to be modified. If we split the watermark up into $L$ bits and use the first approach for detection, $E/N_0$ is reduced by a factor of $L$ for each bit, and we obtain

$$ P_{D_{L \infty}} = \left(1 - Q\left(\frac{E}{\sqrt{L \cdot N_0}}\right)\right)^L $$

(25)

where $SNR$ is the ratio of the total watermark energy to the noise PSD. On the other hand, if we detect this yes/no watermark in the conventional way, we find

$$ P_{D_{c3}} \approx 1 - Q\left(\sqrt{\frac{E}{N_0}} - Q^{-1}(\rho)\right) $$

(26)

The false alarm probabilities are independent of $M$ and are thus unaffected.