| | |
|---|---|
| **REPORT DOCUMENTATION PAGE** | Form Approved OMB No. 0704-0188 |

| | | |
|---|---|---|
| **1. REPORT DATE** ***(DD-MM-YYYY)***<br>04-04-2005 | **2. REPORT TYPE**<br>Final Report | **3. DATES COVERED** *(From – To)*<br>7 May 2003 - 07-Jan-05 |

| | |
|---|---|
| **4. TITLE AND SUBTITLE**<br>Digital Watermarking Using Syndrome Codes | **5a. CONTRACT NUMBER**<br>FA8655-03-1-3045 |
| | **5b. GRANT NUMBER** |
| | **5c. PROGRAM ELEMENT NUMBER** |
| **6. AUTHOR(S)**<br>Dr. Ingemar J Cox | **5d. PROJECT NUMBER** |
| | **5d. TASK NUMBER** |
| | **5e. WORK UNIT NUMBER** |

| | |
|---|---|
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**<br>University College London<br>Torrington Place<br>London WC1E 7JE<br>United Kingdom | **8. PERFORMING ORGANIZATION REPORT NUMBER**<br>N/A |
| **9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**<br>EOARD<br>PSC 821 BOX 14<br>FPO AE 09421-0014 | **10. SPONSOR/MONITOR'S ACRONYM(S)** |
| | **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**<br>Grant 03-3045 |

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

The Grantee will conduct research on 'informed watermarking.' Recent analyses proved that the theoretical limit to digital watermarking is much higher than previously believed, however no practical means for accomplishing the theoretical rates are available. As detailed in the technical proposal, the PI will investigate means to achieve increased embedding rates. Specifically, he will:
1. refine the trellis-based coding techniques including (i) studying how variations in the trellis architecture (i.e. number or states and arcs) affect performance and (ii) investigating the performance of syndrome coding as a function of the underlying error correction code.
2. investigate approximations to the informed embedding optimizations, including (i) studying how such optimizations can be efficiently computed and (ii) examining the choice of symbol alphabet (binary versus n-symbol alphabets) and their corresponding modulation codes.

Pending satisfactory results of the initial work and availability of funding, additional research would be performed to:
3. Improve the perceptual modeling required to hide a high data rate embedded signal in a video
4. conduct tests to measure and subsequently model the distortions that occur during the broadcasting of television
5. conduct tests to determine the robustness of our watermarking algorithms to commercial broadcasting.

**15. SUBJECT TERMS**

EOARD, Communications, Digital Watermarking

| | | | | | |
|---|---|---|---|---|---|
| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT**<br>UL | **18, NUMBER OF PAGES**<br>11 | **19a. NAME OF RESPONSIBLE PERSON**<br>PAUL LOSIEWICZ, Ph. D. |
| **a. REPORT**<br>UNCLAS | **b. ABSTRACT**<br>UNCLAS | **c. THIS PAGE**<br>UNCLAS | | | **19b. TELEPHONE NUMBER** *(Include area code)*<br>+44 20 7514 4474 |

# Digital Watermarking Using Syndrome codes

## *Introduction*

Syndrome codes are based on a subtle modification of traditional error correcting codes, and were first introduced by Pradhan et al. [1].

If we have a message “1 0 1 0”, this may be coded using an error correcting code as “0001 1100 1111 1110”. Note that every 4th bit of the ECC code yields the message. We call these bits “base” bits and the remaining bits parity bits.

The syndrome of a received bit sequence provides a measure of the errors present. Suppose we receive the bit sequence “0001 1110 1111 1111” The “base bits” are “1011”

The ECC for these base bits is “0001 1100 1111 1101”. The parity bits are therefore “000 110 111 110”

If we compare the parity bits of the received sequence “000 111 111 111” and the generated sequence “000 110 111 110”, then exclusive OR’ing the two sequences gives “000 001 000 001”. This is the syndrome of the received signal.

Note that the syndrome does not tell us how to correct the errors. The errors may be due to an error in the parity bit or an error in the base bit.

For watermarking, the message is coded in the syndrome of the bit sequence. Thus, the sequence “0001 1100 1111 1110” represents the message “000 000 000 000”. The sequence “0001 1110 1111 1111” represents the message “000 001 000 001”

There are many ways to represent the same message Thus, syndrome codes are a form of “dirty paper code”. We can use a modified trellis to find the bit sequence closest to that of the image and then embed this code word in the image

## *Our Approach*

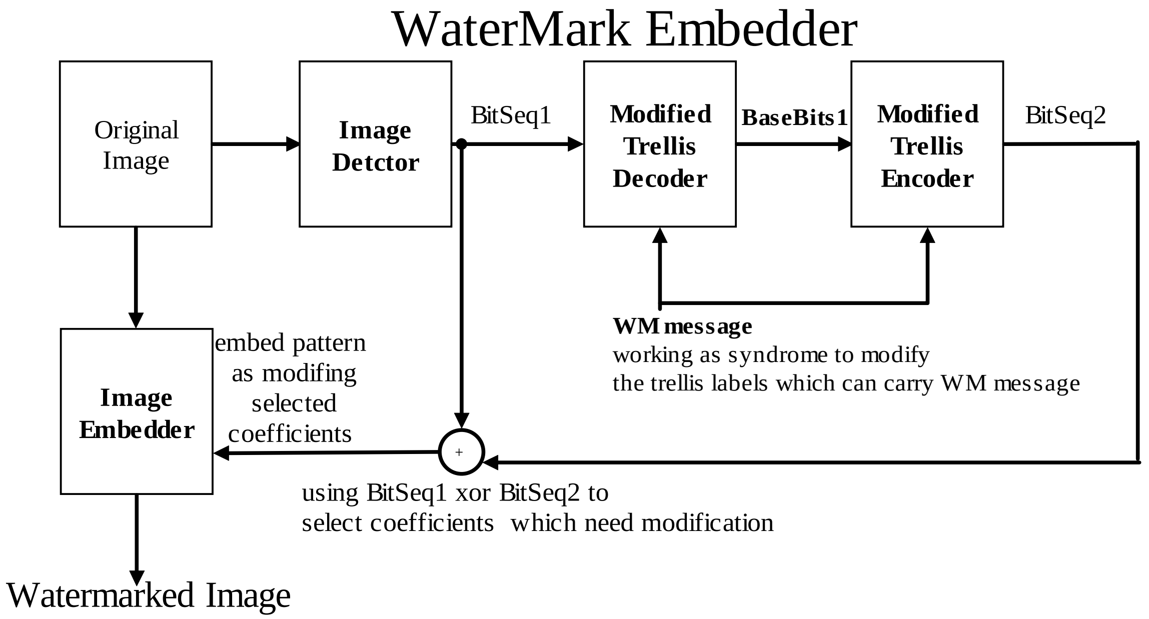
WaterMark Embedder
Original Image
Image Detctor
BitSeq1
Modified Trellis Decoder
BaseBits1
Modified Trellis Encoder
BitSeq2
WM message
working as syndrome to modify
the trellis labels which can carry WM message
Image Embedder
embed pattern as modifing selected coefficients
+
using BitSeq1 xor BitSeq2 to
select coefficients which need modification
Watermarked Image

**Figure 1. Watermark Embedder**

In our approach, we first extract the bit sequence BitSeq1 from the original unwatermarked image. This is a random sequence of bits that are latent in an image. The extraction process can, for example, be based on extracting signals from the original pixels or coefficients in the DCT domain, and then correlating each short sub-signal against prescribed vectors. If the correlation is positive then a “1” is encoded, otherwise a “0”.

Figure 1 schematically shows the processing procedure when embedding.

The embedding process is:

1) Extract vectors from original image and compute the correlation with prescribed reference patterns to obtain bit sequence BitSeq1.

2) Modify the trellis labels according to the watermark message to be embedded. This modification ensures that the syndrome errors encode the message.

3) Run the modified trellis decoder to obtain the base bit sequence BaseBits1.

4) Run the modified trellis encoder with BaseBits1 as the input to obtain the bit sequence BitSeq2. The syndrome of this bit sequence correctly encodes the message.

5) Exclusive OR Bitseq1 with Bitseq2 to determine which bits in the unwatermarked image must be modified. Modify these bits accordingly.

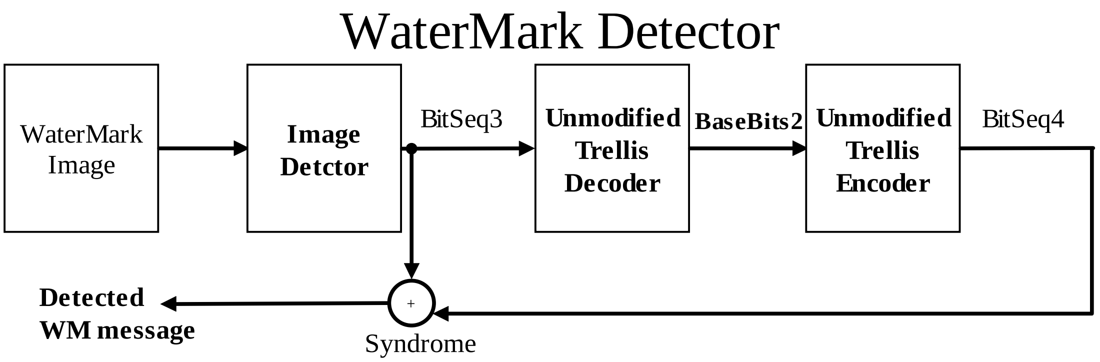

**Figure 2. Watermark Detector**

The diagram of watermark Detector is shown in Figure 2. The detection process is:

1. Extract the bits from the watermarked image to obtain bit sequence BitSeq3
2. Apply the unmodified trellis decoder to BitSeq3 to obtain the base bits BaseBits2
3. Re-ecode the base bits with the unmodified trellis encoder to obtain bit sequence BitSeq4.
4. Exclusive OR BitSeq3 with BitSeq4 to obtain the syndrome, i.e. the message

## *Problems*

If we modified the trellis too much, i.e. introduce too many errors in the syndrome, then we can exceed the error correcting capacity of the code. In this case, we cannot extract the correct Base bits (BaseBits2) from BitSeq2 and the message contains errors. One way to solve this is to directly extract the base bits from BitSeq3 rather than using traditional Trellis Decoder. However, then, the base bits of BitSeq3 are privileged, i.e. we are much more susceptible to errors in these bits. We are investigating this problem.

## *Bibliography*

[1] Sandeep Pradhan and K. Ramchandran, Distributed Source Coding Using Syndromes: Design and Construction, Proc. Of the Data Compression, 158-167, March 1999

[2] M. Costa. Writing on dirty paper. *IEEE Trans. on Information Theory*, 29:439–441, May 1983.

[3] J. Chou,S. Sandeep Pradhan and K. Ramchandran,On the Duality between Distributed Source Coding and Data Hiding, IEEE Conf. on acoustics, speech and signal processing, 2061-2064,1999.

## 1. *Abstract*

The informed embedding algorithm of [1] is computationally expensive due the to its iterative structure. In February, we described preliminary results of an algorithm that embeds a given codeword based on a global estimate of robustness. While the computational time of this algorithm was considerably less, it is clearly suboptimal embedding region, exhibiting a large perceptual distortion in order to reach a low, almost zero, Bit Error Rate (BER).

Here, we describe a *local* algorithm for informed embedding. That is, given the codeword to be embedded, each of the L-bits of the codeword is embedded independently. Both the computational cost and the fidelity are improved, but still remain inferior to the original algorithm.

## 2. *Introduction*

We use a 64 state and 64 arcs per state trellis, as described in [1]. The number of bit we embed is L = 1380. Each path through the trellis codes an L-bit message as a length L × N-dimensional vector that is the concatenation of the L labels, or reference patterns, associated with each arc. The dimension of these reference patterns is N = 12. We use the Viterbi algorithm to identify the path, through the trellis whose L × N reference vector has the highest correlation with the extracted vector.

## 3. *Local informed embedding algorithm*

As before, the trellis structure defines allowable codewords. These codewords are zero mean and uniform variance. They can consequently be regarded as points distributed uniformly on the surface of a high dimensional sphere.

$$\overline{w} = 0\,,\ \ |w| = 1$$

During detection, we want to find the codeword that has the highest correlation with the Work to be tested. It is equivalent to find the codeword with the highest correlation coefficients.

$$Z_{nc} = \underset{w}{Max}\frac{c \cdot w}{|c|}$$

This can be efficiently performed using Viterbi decoding.

Any codeword we need to embed consists of L-bits and each bit has an associated reference pattern. These local reference patterns correspond to a linear subspace of the global marking space. Thus, for each bit of the codeword, we draw a cone around it in the bit's subspace, and embed into that cone with the informed embedding method described in [2].

The local informed embedding algorithm proceeds as follows for each bit *i* of the codeword:

1. Convert the image into 8×8 block DCT domain, and extract the 12 lowest frequency AC terms of the $i$ th block into a vector, $v_o^i$.

2. Project $v_o^i$ into the $w_a^i$ plane, where $w_a^i$ is the reference pattern associated with the I-th bit. The two-dimensional plane that contains both $v_o^i$ and $w_a^i$ is described by two orthogonal unit vectors X and Y, obtained by Gram-Schmidt ortho-normalization.

3. Find the point in the embedding region $< x_{vw}, y_{vw} >$, that is closest to $< x_{vo}, y_{vo} >$.

$$x_{vw} = tcc \times \sqrt{\frac{R + y_{vw}^2}{1 - tcc^2}}$$

4. Project $< x_{vw}, y_{vw} >$ back to 12-dimensional space to obtain a watermarked vector, $v_w^i$.

5. Invert the $v_w$ back to 8×8 block DCT domain, and back to spatial domain, to obtain the watermarked block.

The above algorithm takes approximately 3 seconds to watermark an image as opposed to the 20 minutes required of the original iterative algorithm.

## Experimental results

We applied the algorithm to 550 images and the resulting average Bit Error Rate versus Watson distance is shown in Figure 1. We see that when Watson distance is around 107, the average BER is approximately 0.01%.

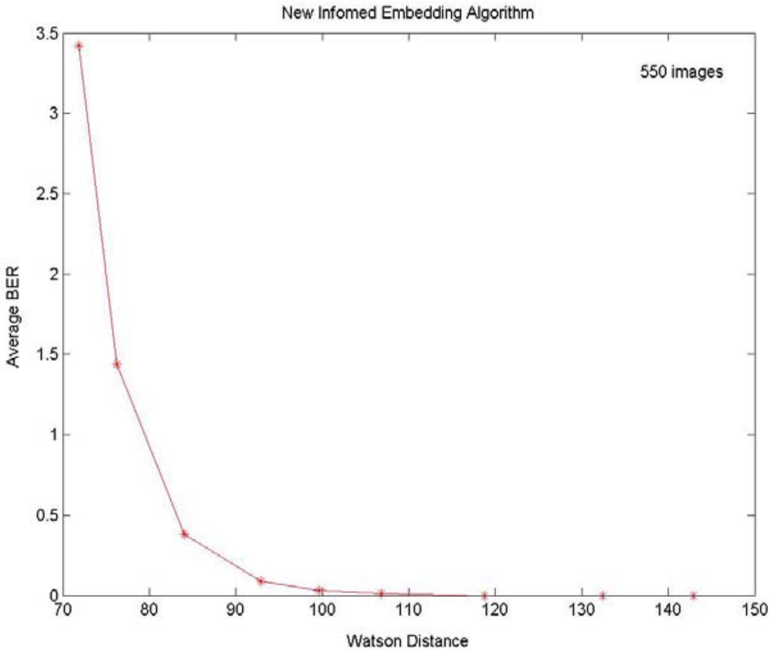

Figure 1: The average Bit Error Rate versus different Watson distance

Table 1 compares the performance of the original iterative algorithm in [1], with the global algorithm we presented in February and new local algorithm described here.
We see that for similar bit and message error rates, the original algorithm has the smallest Watson distance, i.e. it has the least perceptual impact on the images. The new local algorithm has a Watson distance almost twice as great while the earlier global algorithm has a Watson distance almost three times larger than the original algorithm.

The original image and watermarked images from those three algorithms are shown in Figure 3, 4, 5 and 6.

| | Old algorithm | Global algorithm | Local algorithm |
|---|---|---|---|
| Average Watson distance | 77.830771 | 222.242999 | 134.944364 |
| Average BER | 0% | 0.000395% | 0.000264% |
| MER | 0% | 0.545455% | 0.363636% |
| Average Mean Sq. Error | 7.001501 | 108.64804 | 79.440267 |

Table 1 The comparison results

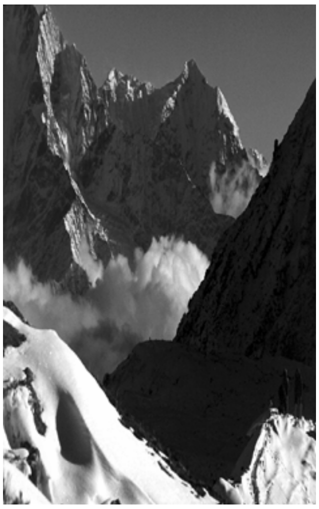

Figure 2 Original Image

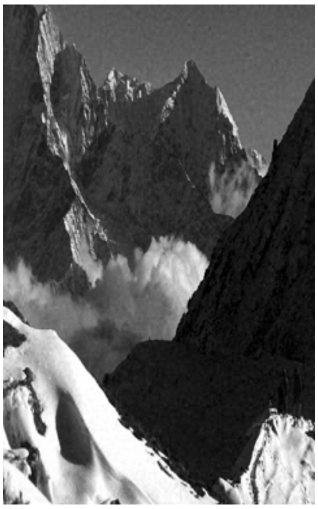

Figure 3 Wm image from old algorithm [1]

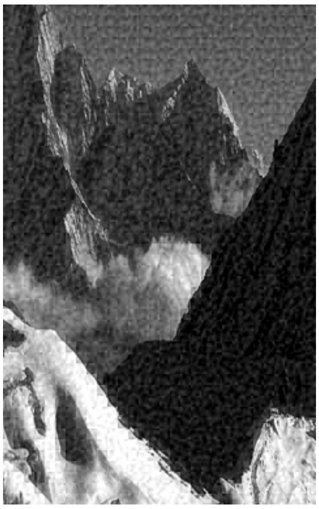

Figure 4 Wm image in global algorithm

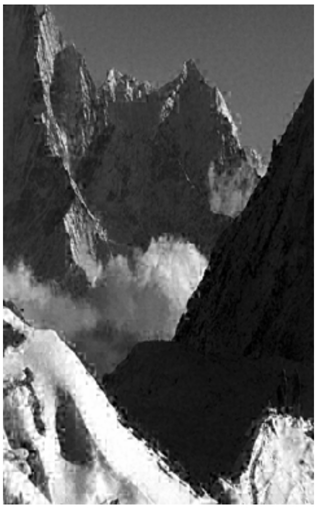

Figure 5 Wm image in local algorithm

Notice in Figure 6, that blocking artifacts are visible along the right diagonal incline of the mountain. Further work is needed to understand how to improve the perceptual quality.

[1] M. L. Miller, G. J. Doerr, and I. J. Cox, “Applying informed coding and embedding to design a robust, high capacity watermark,” IEEE Trans. on Image Processing.

[2] I. J. Cox, M. L. Miller and J. A. Bloom, “Digital Watermarking”, Morgan Kaufmann, 2001.

# Experiment to investigate how performance is affected by the number of states in the trellis

The objective of this experiment is to investigate the effect of varying the number of states on the BER performance. This experiment is done by varying the number of arcs per state (keeping the number of states to be two and the embedding strength to be eight) to examine the bit error rate (BER). The following are the details of the experimental setup.

- The trellis is based on dirty paper codes whereby there are more than two arcs entering and leaving a state. In this experiment, only two states are used.
- 100,000 'images' are used in this experiment. The extracted vectors of these images, denoted by $c_0$, are simulated by normally distributed numbers for every element in the extracted vectors.
- Every arc, in the trellis, is represented by a reference vector. The elements of all reference vectors leaving each state are also normally distributed numbers. For simplicity, the set of reference vectors in the first step of the trellis are repeated for all other steps in the trellis. In this experiment, the number of arcs used varies from 2 to 512.
- A message of 1,000 bits are embedded into an extracted vector of length 1,000 x N, where N is the dimension of the reference vector. The extracted vector is the original image in the marking space. In this experiment, N takes the values of 16, 32 and 64.
- During encoding, the extracted vector goes through the restricted trellis and the path with the highest linear correlation between the extracted vector and the reference vector is identified as $w_r$. Then this reference vector is blindly embedded into the image, i.e. watermarked vector, $c_w = c_0 + \alpha w_r$, where $\alpha$ takes the value of 8.
- During decoding, the watermarked vector goes through the full trellis and the path with the highest linear correlation between the watermarked vectors and the reference vector is identified. The decoded message is compared with the input message and the average bit error rate (BER) is computed.

## Experimental Results

Figure 1 shows the result for a two-state trellis with α = 8 and N = 16, 32, 64. It also includes the result for a one-state trellis with α = 8 and N = 16, 32, 64. Note that the result for one-state trellis is taken from the February report for purposes of comparison. As the number of states is increased, the BER performance has improved. It is unclear how much of the improvement is due to informed coding and how much is dues to the inherent error correction properties of the trellis. This will be investigated.

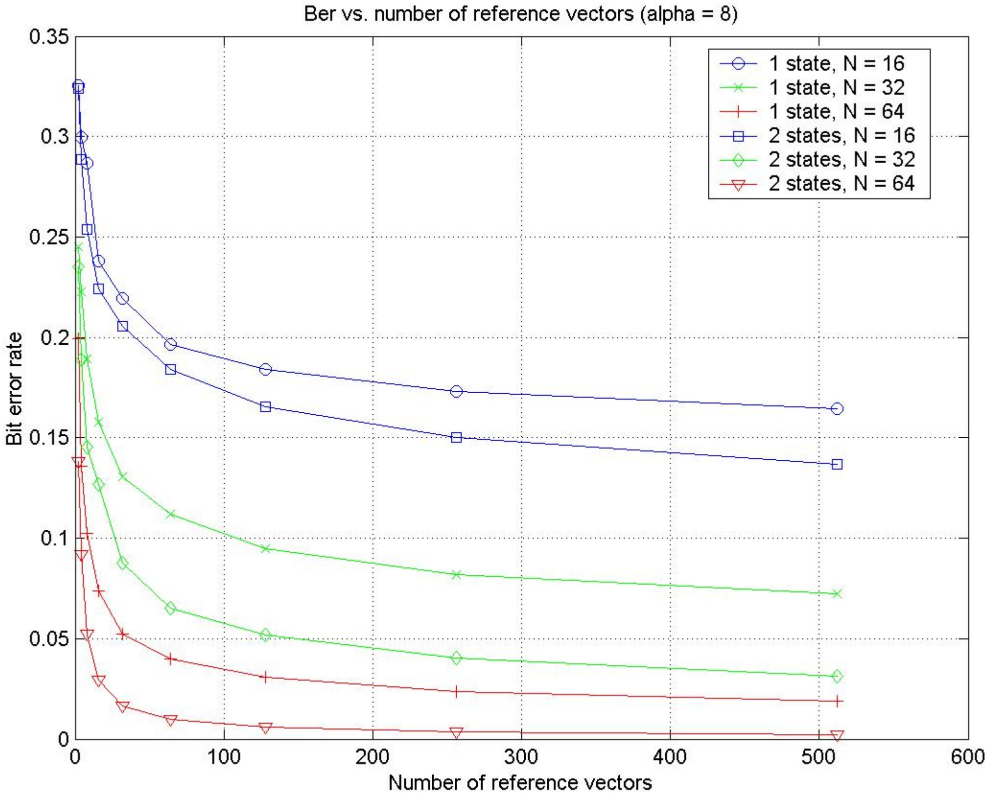

Figure 1: BER vs. number of reference vectors (Number of states = (1, 2), α = 8)