Trellis Coded Modulation to Improve Dirty Paper Trellis Watermarking

Chin Kiong Wang, Gwenaël Doërr and Ingemar Cox

University College London Adastral Park
Ross Building 2, Martlesham IP5 3RE, UK
http://www.adastral.ucl.ac.uk

ABSTRACT

Lattice codes and quantization-based techniques have received considerable attention as a method of digital watermarking with side information. However, these algorithms are also recognized to be highly sensitive to common signal processing primitives such as valuemetric scaling, e.g. changes in volume of a song, or re-quantization, e.g. JPEG compression. Hence, it is reasonable to investigate alternative codes which may offer the potential for superior performance. In previous work, we designed a new class of codes, called dirty paper trellis codes (DPTC), which are by construction immune to valuemetric scaling. Although DPTC have shown themselves to be resistant to certain classes attacks, the trellises were generated randomly, thus leading to suboptimal codes. In this paper, Ungerboeck's work on trellis coded modulation (TCM) is exploited to significantly improve the performance of the DPTC. Experimental results show that the resulting trellises significantly outperform the original design.

Keywords: Watermarking, informed coding, trellis coded modulation

1. INTRODUCTION

Digital watermarking[1] consists of imperceptibly altering multimedia content, such as a movie or song, to convey some information. This process is done in such a way that the hidden data is not perceptible to an observer even though it can be detected by a computer. Modeling watermarking as communications with side information has recently triggered significant advances. As depicted in Figure 1, it refers to communicating over a channel in which the transmitter has additional knowledge or side information about the channel. Theoretical studies of communications channel with two noise sources, one of which is completely known to the transmitter, but neither of which is known to the receiver, revealed that, under some Gaussian assumptions, the channel capacity was equivalent to a channel in which the first (known) noise source was absent.[2,3] From a watermarking perspective, this first noise source is equivalent to the cover work $\mathbf{c}_o$, e.g. an image, a song , a movie, etc. This implies that the cover Work need not interfere with the embedded watermark and thus offers the potential to hide a greater number of bits.[4–6]

In practice, this is done by using dirty paper codes which associate several codewords to a single message. Three main techniques have been proposed for watermarking with side informations. These are lattice codes,[7] syndrome codes[6] and dirty paper trellis codes (DPTC).[8] Up to now, lattice codes, also referred to as quantization index modulation (QIM), have received most attention due to their ease of implementation and their low computational cost. Nevertheless, they are usually criticized for being highly sensitive to valuemetric scaling, e.g. changes in brightness of an image or changes in volume of a song, which are very common signal processing primitives. Recently, there has been significant progress toward resolving this issue.[9–13] However, other limitations remain, including sensitivity to re-quantization, such as JPEG compression.[14,15]

An efficient way to overcome sensitivity to valuemetric scaling is to use codewords that are located on the surface of a multi-dimensional sphere, in contrast with QIM where codewords are located on a multi-dimensional lattice. In other words, this alternative approach considers the whole cover Work as a vector whose *direction* is quantized. In practice, this reduces to defining points on a multi-dimensional sphere i.e. designing a spherical

Send correspondence to Gwenaël Doërr: E-mail: g.doerr@adastral.ucl.ac.uk, Telephone: +44 (0)1473 663 720

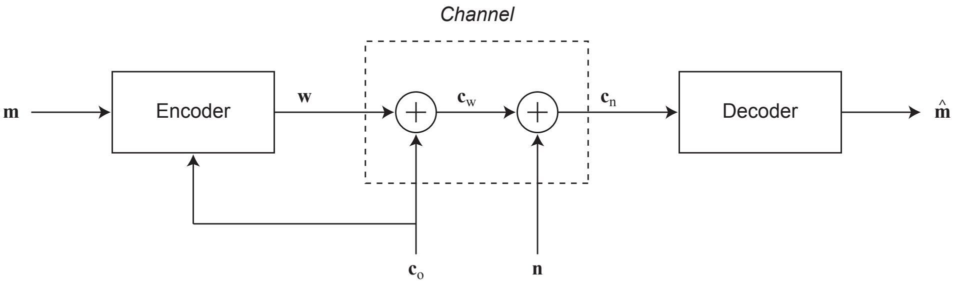

Figure 1. Channel coding with side information about the channel at the encoder: $\mathbf{c}_\mathrm{o} \sim \mathcal{N}(0, \sigma_c)$, $\mathbf{n} \sim \mathcal{N}(0, \sigma_n)$ and $\mathrm{E}[\mathbf{w}^2] \leq d$. The capacity of this channel is identical to that of an AWGN channel having noise component $\mathbf{n}$ only.

codebook. Although this approach was proposed in,[16] there has been very little subsequent research in this direction.[8, 17, 18] The key point then is to define *good* spherical codes. The codewords should be well separated and uniformly distributed throughout the surface of the n-sphere. This ensures that (i) the codewords are robust to noise and that (ii) the distortion to the content is minimized.

In,[18] the codewords are obtained by quantizing a set of angles which defines the direction of any vector in the multidimensional space e.g. the longitude and latitude in 3-D. This leads to a suboptimal distribution of codewords: in 3-D, there is a concentration of codewords at the poles. In contrast, with a good spherical code, the minimum distance between codewords should be maximized, thus leading to a regular distribution of the codewords over the surface of the sphere.[19]

In Section 2, the original design of DPTC is summarized. However, it is now apparent that the random generation of codewords is far from optimum. To address this issue, trellis coded modulation (TCM)[20, 21] is investigated in Section 3 to further improve the performance of DPTC. In Section 4, experimental results are reported which demonstrate that the modified DPTC design clearly outperform the original one. Finally, conclusions are discussed in Section 5.

2. WATERMARKING USING DIRTY PAPER TRELLIS CODES

Before describing DPTC, it is helpful to consider conventional trellis codes. Trellis coding is an efficient method to code and decode a message using an error correcting code. A conventional trellis is depicted in Figure 2. Each node has two arcs emanating from it to two different nodes in the next column of nodes. A step is defined to be the transition from one column of nodes to the next column of nodes, moving from left to right. Each step corresponds to one message bit and each arc is labeled with a reference pattern of length N i.e. a N-sample pseudo-random sequence. Starting from node A0, the trellis is traversed from left to right by choosing a bold arc if the message bit is '1' or an non-bold arc if the bit is '0'. Thus, each L-bit message m is mapped to a unique L-step path through the trellis and the output codeword $\mathbf{w}_m$ is obtained by concatenating the reference patterns associated with the arcs of the path. This one-to-one mapping between the input message m and the output codeword $\mathbf{w}_m$ is referred to as *blind* coding by the watermarking community. This is to emphasize the fact that the coding process does not make use of any available side information, e.g. the original cover Work $\mathbf{c}_\mathrm{o}$.

For informed coding, a given message m may be represented by several alternative codewords, one of which is chosen based on some criterion. Thus there is a one-to-many mapping between messages and codewords. An efficient way to map a message to a desired codeword, consists of modifying a conventional trellis so that more than two arcs leave and enter a node. Such a trellis is shown in Figure 3 and is referred to as dirty paper trellis.[22] A dirty paper trellis has the property that several paths through the trellis encode the same message. It is consequently necessary to tailor a procedure which decides which path, and by extension which watermark signal, will be retained for embedding. This decision is based on the original cover Work $\mathbf{c}_\mathrm{o}$.

First, the dirty paper trellis is simplified so that all paths through the trellis, and thus all the associated codewords, encode the desired message m. This is accomplished by removing all the arcs which do not encode

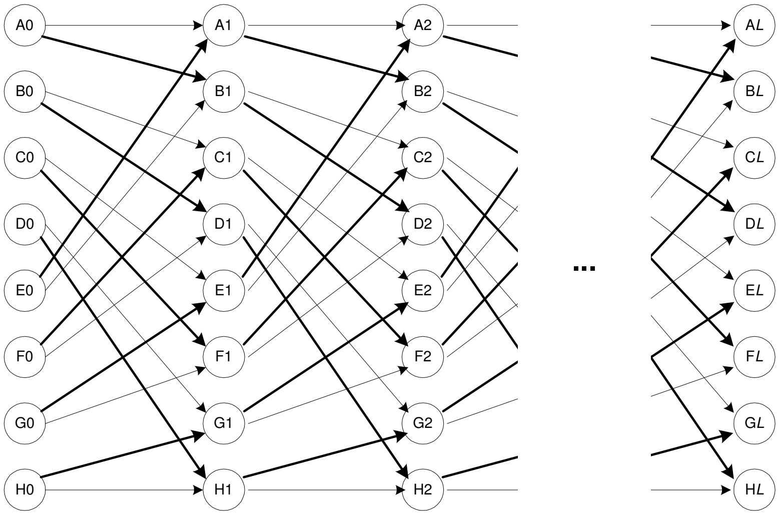

Figure 2. Conventional 8-state trellis: two arcs enter/leave from each state, one encoding a '1' (bold arc) and the other a '0' (non-bold arc). Hence, for a given message m, there is only a single path through the trellis which encodes the desired message. Side information, i.e. the cover Work, has no influence on the message encoding process.

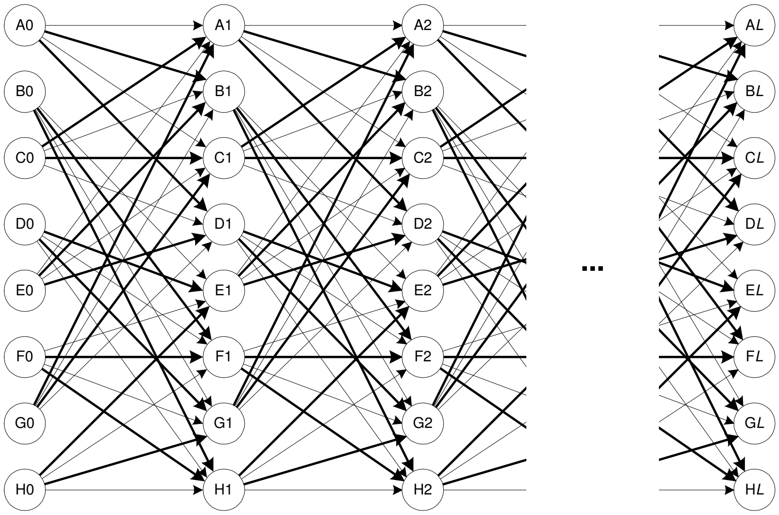

Figure 3. Dirty paper 8-states trellis: more than two arcs enter/leave from each state, half of them encoding a '1' (bold arc) and the other half a '0' (non-bold arc). Hence, there are alternative paths through the trellis which encode the same message. Depending on the available side information, i.e. the cover Work, one of these paths will be selected for embedding.

the correct bits. For example, if the first message bit is a '0', the bold arcs are removed in the first step (nodes A0 . . . H0 to nodes A1 . . . H1). A Viterbi decoder[23] is then run to find the codeword which has the highest linear correlation with the cover Work $\mathbf{c}_\mathrm{o}$. To do so, each step of the trellis is associated with a portion of the cover Work $\mathbf{c}_\mathrm{o}$ e.g. a few samples or coefficients. The cost of traversing an arc is then defined as the linear correlation between the reference pattern associated with the arc and the relevant portion of the cover Work. As a result, the Viterbi decoder finds the path which exhibits the highest linear correlation with the cover Work, i.e. the codeword $\mathbf{w}_m$ encoding the desired message m whose direction is the closest to the cover Work $\mathbf{c}_\mathrm{o}$.

Once the codeword to be embedded has been identified, an efficient embedding procedure has to be defined which ensures that the hidden payload m is properly retrieved on the receiver side. Since the detector does not know the hidden message, it is running the Viterbi decoder with the whole dirty paper trellis depicted in Figure 3. Hence, the goal of the embedder is to modify the cover Work $\mathbf{c}_\mathrm{o}$ so that the codeword $\mathbf{w}_m$ is extracted even if distortions occur afterward. In other words, the direction of the cover Work should be modified so that it is closer to $\mathbf{w}_m$ than to any other codeword in the whole dirty paper trellis. Several embedding primitives have been proposed, including iterative algorithms[8, 24] or algorithms exploiting suboptimal embedding regions.[25] More simply, the direction of the cover Work, $\mathbf{c}_\mathrm{o}$ can also be quantized to the selected codeword, i.e. embedding $\mathbf{w}_m$ reduces to:

$$\mathbf{c}_\mathrm{w} = \frac{|\mathbf{c}_\mathrm{o}|}{|\mathbf{w}_m|}\mathbf{w}_m \tag{1}$$

where $\mathbf{c}_\mathrm{w}$ is the resulting watermarked Work. At the receiver, once the path through the trellis which has the highest linear correlation with the watermarked cover Work, $\mathbf{c}_\mathrm{w}$, has been identified, the hidden message is determined by examining each arc in this path to determine whether it encodes a ‘1’ or a ‘0’.

The performances of DPTC watermarking in terms of bit error rate (BER) are tightly related to the distribution of the codewords defined by the structure of the dirty trellis over the surface of the $(N.L)$-dimensional sphere. Several parameters are likely to influence this distribution:

- the number of states in the trellis,
- the number of arcs entering/leaving each state,
- the set of reference patterns used to label the arcs,
- the connectivity between the states,
- the mapping between the reference patterns and the arcs,
- the mapping between the dirty bits (bold vs. non-bold) and the arcs.

Let us ignore for the moment the *dirty paper nature* of the trellis depicted in Figure 3 i.e. the distinction between bold and non-bold arcs is no longer made. As a result, we obtain a trellis with several arcs leaving/entering each state. From a communications perspective, it simply means that more than one bit is emitted per sampling period. Such a framework has been studied by Ungerboeck in the context of trellis coded modulation,[20, 21] which is described in the next section. Significant progress has indeed been made within the communications community to optimize such codes and it is interesting to extend this work to DPTC.

3. TRELLIS CODED MODULATION

In communications, channel capacity curves[20] suggest that doubling the number of emitted symbols can improve the reliability of digital communications over an additive white Gaussian channel by up to 6 dB. In other words, if k bits are to be transmitted, then using an alphabet $\mathcal{A}$ of 2^{k+1} symbols and exploiting an appropriate coding/decoding scheme, it is possible to significantly enhance the robustness of digital communications against additive noise. This may be done by feeding a conventional encoder with rate $R = k/(k+1)$ with the input binary data and then mapping the resulting $(k+1)$-long codewords into the symbol set of the alphabet. This framework was originally proposed by Ungerboeck in 1982[20] and is depicted in Figure 4. This approach is commonly referred to as trellis coded modulation (TCM).

When the emitted sequence of symbols $\{\mathbf{s}_t\}$ is corrupted by additive white Gaussian noise (AWGN), the decoder may start making wrong decisions, possibly engaging in a path in the trellis which differs by more than one transition from the correct path. Such mistakes are called error-events. If soft maximum likelihood decoding, such as Viterbi decoding, is applied, the probability of such errors is generally well approximated at high signal-to-noise ratio (SNR) by:

$$\mathrm{P}_e \cong N_\mathrm{free}.\mathrm{Q}(d_\mathrm{free}/2\sigma) \tag{2}$$

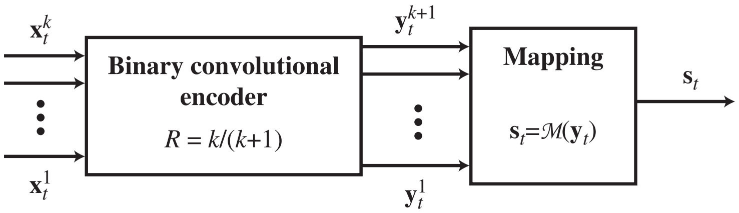

Figure 4. Trellis coded modulation (TCM): a binary convolutional encoder with rate $R = k/(k+1)$ is fed at time t with a k-bits binary word $\mathbf{x}_t$ and a mapping function defines defines which symbol $\mathbf{s}_t$ to emit depending on the $(k+1)$-bits encoded output $\mathbf{y}_t$.

where Q(.) is the Gaussian error integral. The free distance d_{free} of the trellis refers to the minimum Euclidean distance:

$$d_{\text{free}} = \min_{\{\mathbf{s}_t\} \neq \{\mathbf{s}'_t\}} \left[\sum_{t=1}^{L} ||\mathbf{s}_t - \mathbf{s}'_t||^2 \right]^{\frac{1}{2}} \tag{3}$$

between all pairs of sequences $\{\mathbf{s}_t\}$ and $\{\mathbf{s}'_t\}$ that the encoder can produce. On the other hand, N_{free} denotes the average number of error events with distance d_{free}. Looking at Equation (2), it is clear that the trellis should be designed in such a way that the free distance is maximized.

Before going into any optimization process, it is necessary to define the alphabet $\mathcal{A}$ of symbols which will be emitted by the encoder. One desired property is that these symbols are as separated as possible to avoid confusion on the receiver side. Furthermore, since the underlying goal is to obtain a code that is invariant to valuemetric scaling in digital watermarking, equi-energetic symbols, such as phase shift keying (PSK) symbols in 2-D, will be considered in the remainder of this paper, i.e. each arc of the trellis is associated with a 2-sample symbol. However, we note that, for digital watermarking, it is a common practice to use many more samples per arc, e.g. a few tens. In this perspective, previous work by Pietrobon *et al.*[21] is of particular interest. They investigated the use of $L\times$MPSK symbols, i.e. the concatenation of L 2-D MPSK symbols, for TCM which results in a alphabet of M^L symbols of $2L$ dimensions[*].

Once the alphabet of symbols has been defined, TCM iteratively partitions the alphabet into subsets with increasing minimum distance $\Delta_1 < \Delta_2 < \ldots$ between the symbols belonging to these subsets. This process has a critical influence on performance since it is used to map the output encoded bits of the convolutional encoder. This concept is illustrated in Figure 5 for an 8-PSK alphabet. At each partition level, the considered set of symbols is separated into two subsets so that the minimum distance within the subsets increases. At the end, the obtained subset are arranged along a binary tree. Thus it is possible to to label each leaf of this tree with a $\log_2(|\mathcal{A}|)$-bit label. In higher dimensions,[21] it might not be possible to increase the minimum distance at each partition level. In such cases, the partitioning process should lead to a maximum reduction in the number of nearest neighbours within the smaller subsets. In practice, with $L\times$MPSK symbols, it is possible to perform partitioning in a systematic manner by making use of block codes.[21]

The final step of TCM consists in performing an exhaustive search to identify the convolutional code which maximize the free distance d_{free} of the trellis. This is equivalent to enumerating all possible connectivity configurations and retaining the one configuration with the largest free distance, i.e. the maximum distance between two closest codewords as defined in Equation (3). A convolutional code can be represented by a set of parity-check polynomials $\{\mathrm{H}^{k+1}(D) \ldots \mathrm{H}^1(D)\}$. Any binary sequence $\mathbf{y}_t$ output by the convolutional encoder satisfies the

[*] For security reasons, it may be required that the alphabet is not disclosed to the public, i.e. it is necessary to have a secret key K to have access to the alphabet. This can be easily done by applying a pseudo-random rotation to a generic known alphabet such as the one described in.[21] In some sense, it is similar to the dither term introduced in QIM schemes.[7]

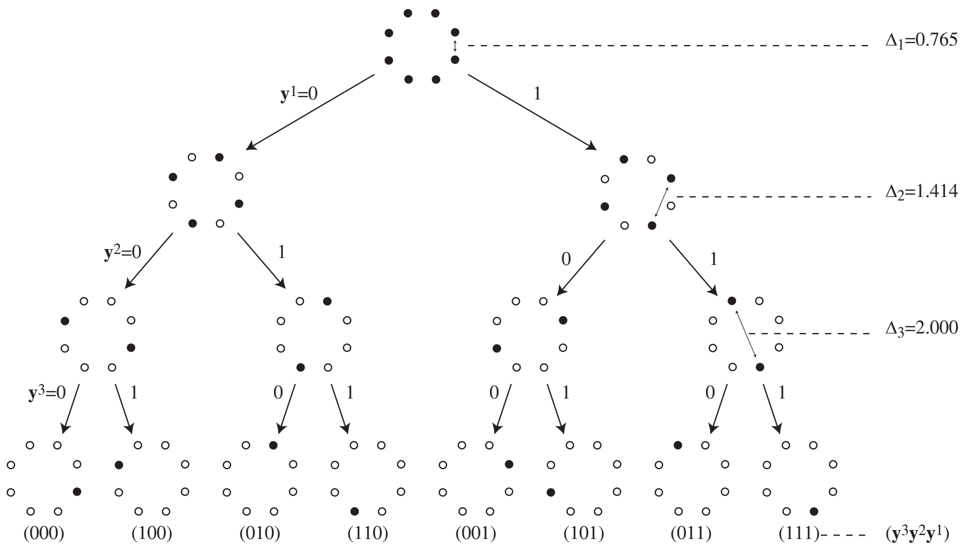

Figure 5. Partitioning of 8-PSK symbols. At each partition level, the considered set of symbols is separated into two so that the minimum distance within the resulting subsets is increased ($\Delta_1 < \Delta_2 < \Delta_3$). This results in a binary tree whose leaves are labeled with a 3-bit label.

parity check equation:

$$[\mathbf{y}_t^{k+1} \dots \mathbf{y}_t^1] \cdot [\mathrm{H}^{k+1}(D) \dots \mathrm{H}^1(D)]^{\mathrm{T}} = 0 \tag{4}$$

where $\mathbf{y}_t^i$ is the i-th bit of the output symbol $\mathbf{y}_t$ at time t, and D is a time delay which indicates that the encoding process is influenced by the previous outputs of the encoder ($D.\mathbf{y}_t = \mathbf{y}_{t-1}$). In other words, each output bit $\mathbf{y}_t^i$ respects a transfer function given by the associated polynomial $\mathrm{H}^i(D)$. The higher is the degree of the polynomial, the farther in the past it is necessary to look to know which bit has to be emitted next.

An efficient way to perform the exhaustive search is to enumerate all possible parity-check polynomials.[20] These polynomials have the following form:

$$\mathrm{H}^j(D) = \sum_{i=0}^{\nu} h_i^j D^i, \quad 1 \leq j \leq k+1 \tag{5}$$

$$\text{with} \quad h_0^j = h_\nu^j = \begin{cases} 0, & j \neq 1 \\ 1, & j = 1 \end{cases}$$

where the h_i^j's are binary coefficients and ν the constraint length of code. The larger ν is, the more *memory* has the code. The exhaustive search then reduces to enumerating all possible combinations of binary coefficients and to keep the set of coefficients which gives the trellis structure with the highest free distance.

The TCM optimization process ensures that the sequences of emitted symbols are as separated as possible. From a watermarking perspective, these sequence of symbols are the codewords $\mathbf{w}_m$ used in Section 2. To use TCM in our DPTC framework, there is still one issue to be addressed: how to assign a dirty paper binary label (bold vs. non-bold) to the arcs of the trellis? For fidelity reasons, it is desired that all the codewords encoding the same message m are well separated, i.e. all the paths through the trellis encoding the message to be hidden should be as separated as possible. With TCM codes, the arcs leaving from each node belong to one of the two subsets at partition level 1. Looking at Figure 5, it means that $\mathbf{y}_t^1 = 0$ (or 1) for all the arcs leaving a given node. These symbols should then be separated into two subsets so that the minimum distance within each subset is maximized. This is exactly what is done for partitioning at level 2. Hence, the bit $\mathbf{y}_t^2$ of the binary symbol output by the convolutional encoder is used as the dirty paper label of each arc.

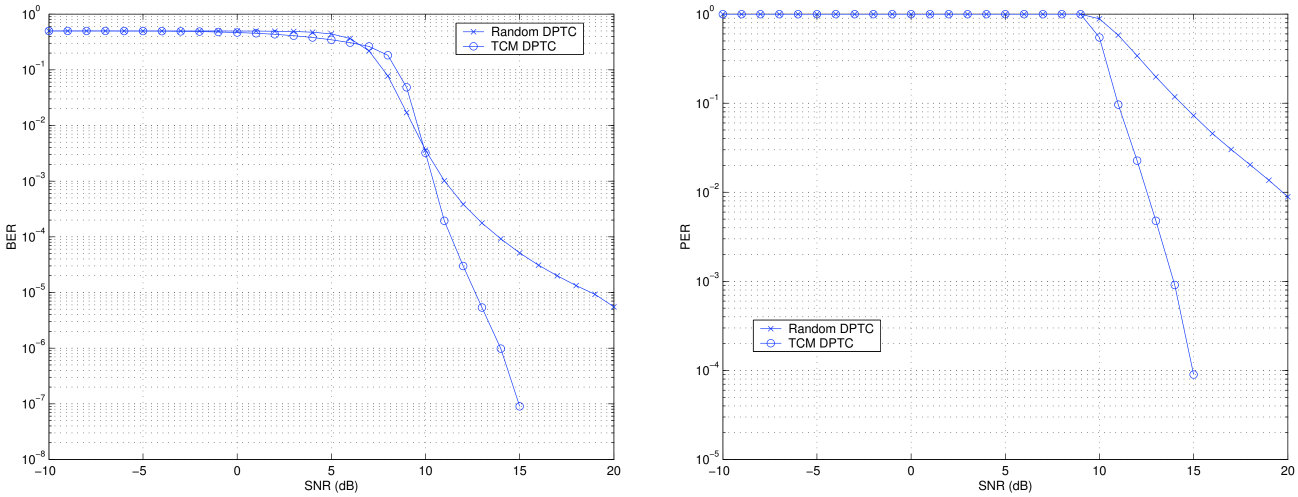

Figure 6. Performance comparison in terms of BER and PER between random and TCM DPTC with synthetic signals and direction quantization embedding.

4. EXPERIMENTAL RESULTS

To verify that the proposed TCM-DPTC outperforms a randomly coded design, experiments are first carried out with synthetic signals. Normally distributed random covers $\mathbf{c}_{\mathrm{o}}$ of length $4L = 4000$, with unit variance, are generated. Next, two kinds of dirty paper trellises with 128 states and 32 arcs per state are generated: either following the original design,[8] or following a TCM-guided design using 2×8PSK symbols based on the Tables VII and XXIV reported in.[21] Random messages of $L = 1000$ bits are then generated and embedded using the direction quantization approach described in Equation (1). Subsequently, the watermarked cover $\mathbf{c}_{\mathrm{w}}$ is corrupted by additive white Gaussian noise with variance σ^2. The signal to noise ratio (SNR) is defined as $SNR = 10.\log(|\mathbf{c}_{\mathrm{w}}|/4L\sigma^2)$. The extracted message obtained by Viterbi decoding is finally compared to the message which has been actually embedded and the bit error rate (BER) is computed. Additionally, the Message Error Rate (MER) and the Path Error Rate (PER) are also computed. The MER relates to the probability of extracting *all* the bits of the message, whereas the PER indicates the probability that the detected path through the trellis is identical to the one chosen for embedding. This experiment is repeated 10000 times to obtain meaningful statistics and for different SNR values.

The obtained results are reported in Figure 6. When considering the BER figure, it is difficult to make any statement in favour or against the TCM design. The two curves cross over which suggests that there is no best design i.e. alternative designs have to be chosen depending on the SNR regime. This is a bit surprising with respect to previous results in digital communications. However, it should be kept in mind that the BER is computed by considering only a single *dirty bit* per symbol, instead of all the symbol bits in digital communications. On the other hand, the PER figure clearly shows that the TCM design always outperforms the random one. For any SNR regime, there is less chance to deviate from the correct path with a TCM DPTC than with a random one. This is tightly related with the fact that the free distance is significantly higher with the TCM design. This was the whole purpose behind the construction of these codes.

In a second round of experiments, a database of 1000 grayscale images of dimension 240×368 has been considered to evaluate the performances of TCM-DPTC under more realistic conditions. First a feature vector is extracted from each image to serve as the cover Work. This is done by computing the 8×8 block discrete cosine transform (DCT) and extracting 4 middle-frequency AC coefficients from each of the $L = 1380$ blocks in the image. This $4L$ vector is then pseudo-randomly shuffled according to the secret key K to obtain the cover Work $\mathbf{c}_{\mathrm{o}}$ which will subsequently be considered for watermarking. Once again, two kinds of dirty paper trellises with 128 states and 32 arcs per state are considered: random DPTC on one hand and TCM-DPTC using 2×8PSK symbols[21] on the other hand. Subsequently, a L-bits long random message is generated and

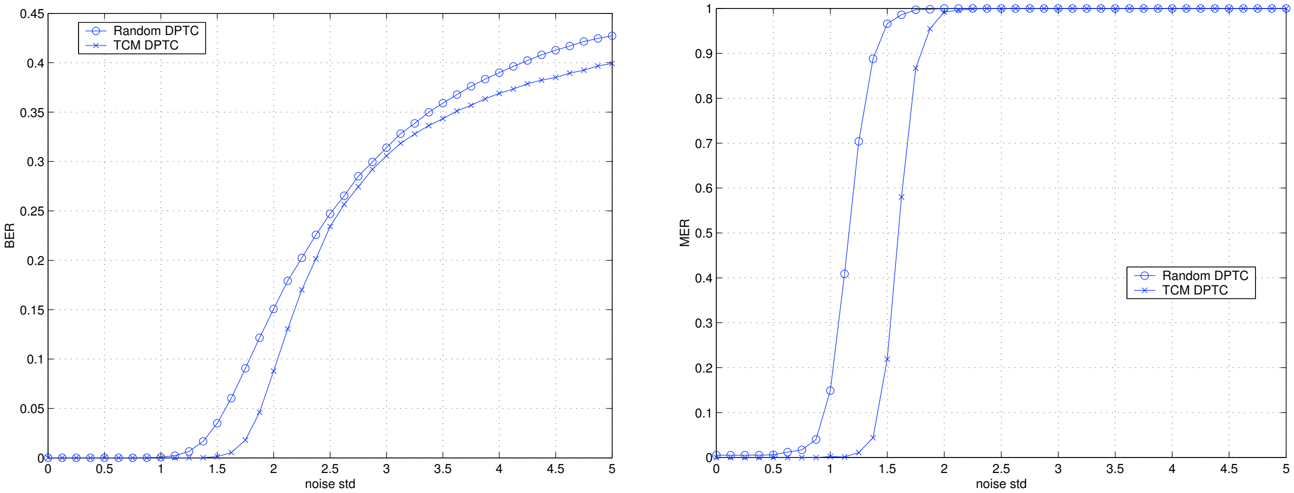

Figure 7. Performances in terms of MER and BER between random and TCM DPTC with real images and iterative embedding .

the obtained codeword $\mathbf{w}_m$ is embedded. In practice, the crude quantization embedding technique given in Equation (1) introduces strong visual distortion. Therefore, the iterative embedding algorithm described in[8] has been preferred. Instead of quantizing the cover Work to the center of the desired detection region, it basically brings it inside with a margin set by a specified robustness parameter. This parameter can be adjusted to control the distortion induced by the watermarking process. In our experiments, this distortion was kept fixed as measured by the mean square error (MSE). The statistics of the observed MSE are reported in Table 1. The distortion has been kept on average around 0.83. However, more importantly, it should be noted that the variance is significantly lower with TCM-DPTC than with random DPTC. Once again, this confirms that TCM induces a better distribution of the codewords on the surface of the sphere.

	Mean	**Variance**
Random DPTC	0.8305	0.3083
TCM-DPTC	0.8295	0.1958

Table 1. Statistics of the MSE with both codes under study.

The watermarked images are then degraded by additive white Gaussian noise with standard deviation σ. As during the embedding process, a cover Work is extracted from each corrupted image. Next, the Viterbi decoder is fed with this work and several numbers are stored e.g. the number of bit errors between the embedded and the extracted payload, the occurrence of a message error or not, the occurrence of a path error or not. These values are then averaged for different images to estimate the BER, MER and PER for a given standard deviation σ. Repeating this process for different value of σ, it is then possible to produce the Figure 7. Once again, looking at the BER plot, the comparison between the two designs is not easy. Although, TCM always outperforms the random approach, the distance between the two curves is fluctuating thus avoiding a clear statement. However, one should keep in mind that with DPTC the error correction and the modulation are combined in a single step. It is not expected that an error correcting code will be added on top of the system. Therefore what is important is whether all the bits of the embedded message have been correctly retrieved or not. When looking at the MER figure, it is clear that the TCM design buys some additional robustness. With comparable embedding distortion, 80% of the TCM-DPTC watermarks survive noise addition up to $\sigma = 1.5$, whereas random DPTC watermarks are only able to cope with noise addition up to $\sigma = 1$. In other words, for a given distortion budget it is possible to go further within the detection region with TCM-DPTC when compared to random DPTC. This is a direct consequence of the better distribution of the codewords.

5. CONCLUSIONS AND PERSPECTIVES

The use of dirty paper codes has received increasing interest within the watermarking community since they are the basis for watermarking based on communications with side information.[2, 3] Although lattices codes and quantization techniques are very popular today,[7] they also have inherent shortcomings such as sensitivity to valuemetric scaling[9–13] and re-quantization.[14, 15] To avoid those pitfalls, the use of spherical codes has been suggested as an alternative. In previous work, a practical implementation using DPTC has been proposed to generate such a code in a high dimensional space.[8] However, the original design was random, thus possibly leading to a suboptimal distribution of the codewords on the unit sphere. Ideally, the codewords should be well separated to optimize the fidelity-robustness watermarking trade-off. In this paper, relevant work on TCM has been revisited to produce better DPTC.[20, 21] Experimental results have validated that TCM-DPTC are more regularly distributed on the unit sphere and consequently ensure higher performances in terms of MER in comparison with random TCM. In the proposed design, each arc was labeled with a concatenation of 2-D MPSK signals as suggested in.[21] Nevertheless, it is not clear whether this approach leads to an optimal distribution of alphabet signals or not and further investigations in this direction will be conducted in the future. On another level, designing *good* spherical codes is also critical when suboptimal embedding techniques are used. For instance in,[25] the detection region is approximated by an hyperbola. The better are distributed the codewords, the less approximated is the true detection region. Finally, as mentioned in the original paper,[20] TCM can be used with lattice alphabets instead of spherical ones. Recent studies have reported that such an approach can significantly improve the performances of traditional quantization-based watermarking schemes.[26]

6. ACKNOWLEDGEMENTS

Gwenaël Doërr's research is supported in part by the Nuffield Fundation through the grant NAL/32707. Ingemar Cox thanks British Telecom (BT) and the Wolfson/Royal Society for their support.

REFERENCES

1. I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, 2001.
2. S. Gel'fand and M. Pinsker, "Coding for channel with random parameters," *Problems of Control and Information Theory* **9**(1), pp. 19–31, 1980.
3. M. Costa, "Writing on dirty paper," *IEEE Transactions on Information Theory* **29**, pp. 439–441, May 1983.
4. B. Chen and G. Wornell, "An information-theoretic approach to the design of robust digital watermarking systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, **IV**, pp. 2061–2064, March 1999.
5. I. Cox, M. Miller, and A. McKellips, "Watermarking as communications with side information," *Proceedings of the IEEE* **87**, pp. 1127–1141, July 1999.
6. J. Chou, S. Pradhan, and K. Ramchandran, "On the duality between distributed source coding and data hiding," in *Proceedings of the 33rd Asilomar Conference on Signals, Systems, and Computers*, **II**, pp. 1503–1507, October 1999.
7. B. Chen and G. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory* **47**, pp. 1423–1443, May 2001.
8. M. Miller, G. Doërr, and I. Cox, "Applying informed coding and informed embedding to design a robust, high capacity watermark," *IEEE Transactions on Image Processing* **13**, pp. 792–807, June 2004.
9. K. Lee, D. Kim, T. Kim, and K. Moon, "EM estimation of scale factor for quantization-based audio watermarking," in *Proceedings of the 2nd International Workshop on Digital Watermarking, Lecture Notes in Computer Science* **2939**, pp. 316–327, October 2003.
10. F. Pérez-Gonzàlez, F. Mosquera, M. Barni, and A. Abrardo, "Rational dither modulation: A high-rate data-hiding method invariant to gain attacks," *IEEE Transactions on Signal Processing, Supplement on Secure Media* **53**, pp. 3960–3975, October 2005.

11. Q. Li and I. Cox, "Using perceptual models to improve fidelity and provide invariance to valumetric scaling for quantization index modulation watermarking," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, **II**, pp. 1–4, March 2005.
12. P. Bas, "A quantization watermarking technique robust to linear and non-linear valumetric distortions using a fractal set of floating quantizers," in *Proceedings of the 7th International Workshop on Information Hiding, Lecture Notes in Computer Science* **3727**, pp. 106–117, June 2005.
13. I. Shterev and R. Lagendijk, "Amplitude scale estimation for quantization-based watermarking," *IEEE Transactions on Signal Processing* **54**, p. 11, November 4146–4155.
14. C. Fei, D. Kundur, and R. Kwong, "Analysis and design of watermarking algorithms for improved resistance to compression," *IEEE Transactions on Image Processing* **13**, pp. 126–144, February 2004.
15. F. Pérez-Gonzàlez, P. Comesaña, and F. Balado, "Dither-modulation data hiding with distortion-compensation: Exact performance analysis and an improved detector for JPEG attacks," in *Proceedings of the IEEE International Conference on Image Processing*, **II**, pp. 503–506, September 2003.
16. B. Chen and G. Wornell, "Quantization index modulation methods for digital watermarking and information embedding of multimedia," *Journal of VLSI Signal Processing* **27**, pp. 7–33, February 2001.
17. I. Svalbe, A. Tirkel, and R. van Schyndel, "Discrete angle watermark encoding and recovery," in *Proceedings of the IAPR International Conference on Pattern Recognition*, **IV**, pp. 246–249, August 2000.
18. F. Ourique, V. Licks, R. Jordan, and F. Pérez-González, "Angle QIM: A novel watermark embedding scheme robust against amplitude scaling distortions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, **II**, pp. 797–800, March 2005.
19. J. Hamkins, *Design and Analysis of Spherical Codes.* PhD thesis, University of Illinois at Urbana-Champaign, USA, September 1996.
20. G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory* **IT-28**, pp. 55–67, January 1982.
21. S. Pietrobon, R. Deng, A. Lafanechére, G. Ungerboeck, and D. Costello, "Trellis-coded multidimensional phase modulation," *IEEE Transactions on Information Theory* **36**, pp. 63–89, January 1990.
22. M. Miller, G. Doërr, and I. Cox, "Dirty paper trellis codes for watermarking," in *Proceedings of the IEEE International Conference on Image Processing*, **II**, pp. 129–132, September 2002.
23. A. Viterbi, *CDMA: Principles of Spread Spectrum Communications*, Addison-Wesley, 1995.
24. M. Miller, "Watermark embedding for black-box channels," in *Proceedings of the 2nd International Workshop on Digital Watermarking, Lecture Notes in Computer Science* **2939**, pp. 18–34, October 2003.
25. L. Lin, I. Cox, G. Doërr, and M. Miller, "An efficient algorithm for informed embedding of dirty paper trellis codes for watermarking," in *Proceedings of the IEEE International Conference on Image Processing*, **I**, pp. 697–700, September 2005.
26. A. Abrardo, M. Barni, F. Pérez-Gonzàlez, and C. Mosquera, "Trellis-coded rational dither modulation for digital watermarking," in *Proceedings of the 4th International Workshop on Digital Watermarking, LNCS* **3710**, pp. 351–360, September 2005.