International Journal of Image Processing and Visual Communication
ISSN (Online) 2319-1724 : Volume 1 , Issue 2 , October 2012

Rotation Based Acceleration of Informed Embedding in DPTC Watermarking Scheme

Marc Chaumont [a,b,c]
(a) Université de Nîmes, F-30021 Nîmes Cedex 1, France
(b) Université de Montpellier 2, UMR5506-LIRMM, F-34095 Montpellier Cedex 5, France
(c) CNRS, UMR5506-LIRMM, F-34392 Montpellier Cedex 5, France
Marc.Chaumont@lirmm.fr

***Abstract*—The Dirty Paper Trellis Code (DPTC) watermarking scheme [1] is a very efficient high rate scheme. It has however a major drawback: its computational complexity. This problem is addressed by using a faster embedding technique. The embedding space is built by projecting some wavelet coefficients onto secret carriers. The fast embedding is achieved with a dichotomous rotation in the Cox, Miller, and Bloom plane. Besides, a modern watermarking scheme should manage the psychovisual impact due to the watermarking signal. This is addressed by using a psychovisual mask. Our low complexity watermarking scheme is compared to two other psychovisual low complexity approaches and results show a good behavior in terms of robustness. The obtained results give a very clear vision, in realistic conditions of use, of the current state-of-the-art for high-rate watermarking schemes of low complexity.**

***Index Terms*—Watermarking, Dirty-Paper Trellis Codes, high rate, informed embedding, robustness, rotation-based embedding, psychovisual watermarking, low complexity.**

I. Introduction

The digital watermarking studies started in the early nineties. In 1998, Costa work is rediscovered [2] and a "new generation" of watermarking schemes, the *informed* watermarking schemes, has been proposed [1], [3]–[7],... Those informed approaches generally outperform the previous non side-informed approaches. With informed approaches, the message is coded by taking into account the *host signal*. In practice, this reduces the interference due to the *host signal* and thus increases the channel capacity [2].

We may broadly define two categories of multi-bit informed watermarking systems: schemes based on lattice codes, more commonly known as quantized based codes (DC-QIM [3], SCS [5], ...) and schemes based on trellis (DPTC [1]). In this paper we address the trellis codes whose original approach [1] is known for its robustness and its high embedding payload. Nevertheless, Dirty Paper Trellis Code (DPTC) has a major weakness: the embedding step uses a Monte Carlo approach which is computationally complex. We propose, in this paper a less complex DPTC. In order to reduce the complexity we propose a rotation based approach in the Cox, Miller, and Bloom plane (abbr. MCB). Besides, we include a generic solution in order to use a psychovisual mask. We also propose by means of experiments on robustness, a better comprehension of the current state-of-the-art for high-rate watermarking schemes of low complexity. Moreover, the obtained conclusions provide additional knowledge to the community and are interesting for practitioners.

Compared to the original DPTC [1], we use a wavelet domain; There is thus no more "block artifacts" (the DCT domain is used in the original DPTC approach), we make difficult the attack presented in [8], since we perform the embedding in a high dimension secret space, we increase the robustness and reduce the distortion by increasing the size of the trellis (as shown [9]); This is made in practice possible thanks to the embedding space and because our technique is fast, we propose a dichotomous rotation in the Cox, Miller, and Bloom (*abbr.* MCB) plane [10] in order to rapidly embed the message codeword, we propose a general solution in order to use any psychovisual mask.

In Section II we re-introduce the concepts of *informed coding* and *informed embedding*. This section presents the original DPTC concepts [1].

In Section III, first, we detail the interesting properties of the embedding space, second, we present the embedding approach, third, we give a solution in order to use a psychovisual mask, and fourth, we have a discussion on security aspects.

In Section IV, four different attacks at various powers are tested on 100 different 256×256 8-bits grey-levels images.

In Section V we conclude and give some perspectives on DTPC.

II. The original DPTC

The general scheme of DPTC is shown in Fig. 1.

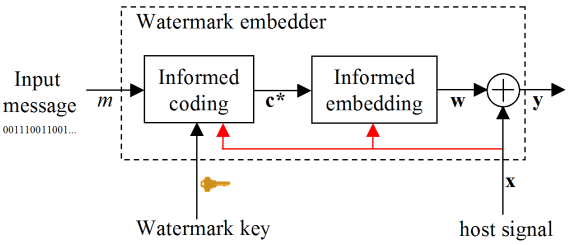


Fig. 1. General watermarking scheme of DPTC [1].

The first step is the image DCT transform in order to obtain the *host signal* $\mathbf{x}$.

The second step is the *informed coding*. The input message m is coded into a codeword $\mathbf{c}^*$ by taking into account the *host signal* $\mathbf{x}$. To perform this encoding, a non-deterministic trellis and the Viterbi algorithm are used.

The last step of the DPTC scheme is the *informed embedding*. It consists in modifying the *host signal* **x** in order to "displace it" in the Voronoï region of the codeword $\mathbf{c}^*$. The displacement vector is named the watermark signal **w**. The addition of the *host signal* **x** and the watermark signal **w** gives the watermarked signal **y**.

Let's now define more precisely the trellis structure, the *informed coding* and the *informed embedding*.

A. Trellis structure

Convolutional codes are a famous form of error correcting codes. For those codes, a states-machine represents the possible transitions given inputs source sequences. Fig. 2 shows a states-machine with four states. One input bit causes a transition to a new state and outputs two bits. The states diagram can also be represented as it evolves in time with a trellis diagram. Fig. 3 shows the trellis associated to the states-machine of Fig. 2. Usually, a trellis is built by placing all the states in column and each possible transitions are drawn by an arc between states at t time and states at $t+1$ time. By convention, bold arcs represent the 1 inputs and non-bold arcs the 0 inputs.

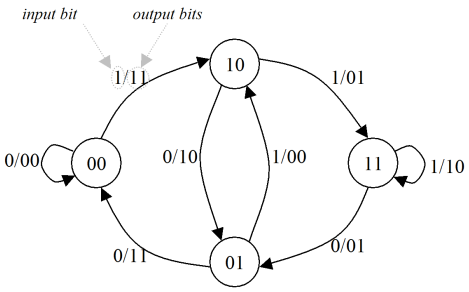


Fig. 2. Binary convolution code's states-machine with 4 states.

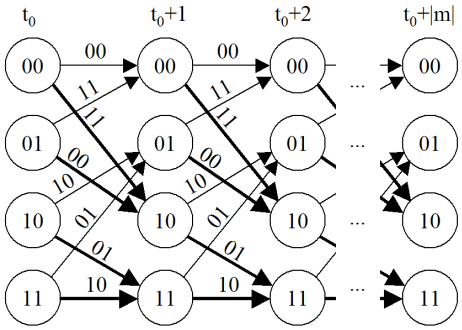


Fig. 3. Binary convolution code's trellis with 4 states.

A convolutional coder takes a sequence of bits in input and generates an output sequence thanks to the states-machine. The output sequence i.e. the codeword is then transmit on a network or stored. The decoder, when receiving the degraded codeword, finds the closest codeword and returns the input sequence related to that codeword. In order to find this closest codeword, a Viterbi algorithm is often used [11]. The principle of the Viterbi algorithm is the same as the Dijsktra shortest path algorithm [12] but is adapted to the very structured form of the trellis. Instead of exploring all the possible paths of the trellis, in order to find the closest codeword, the Viterbi algorithm solves dynamically the problem by keeping the best sub-path at each state at each given time. This way the algorithm does not have to keep track of all possible sub-paths but only one sub-path per state.

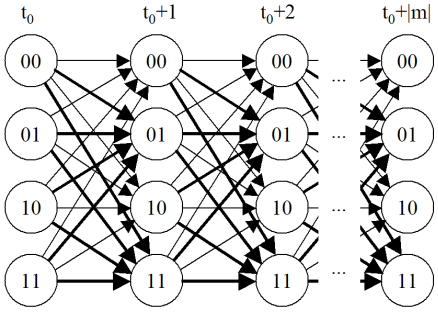


Fig. 4. Dirty paper code's trellis with 4 states and 4 arcs per state.

In DPTC [1], a special trellis is used. In this trellis, each state owns multiple possible transitions given an input bit. Each transition generates output real coefficients. In Fig. 4 we give an example of a trellis with 4 states and 4 arcs per state. With this trellis, an input sequence owns multiple possible output codewords (a codeword is the result of the concatenation of outputs coefficients) since for each state there are multiple possible transitions for the same input bit. An input sequence may thus be coded with different codewords.

In the original DPTC algorithm, the trellis owns 64 states, 64 arcs per state and there are $N_{arc} = 12$ real coefficients pseudo-randomly generated as output arcs values. A bit from the message will be carried by (spread on) $N_{arc} = 12$ coefficients of the host signal **x**. Note that the $N_{arc} = 12$ value is chosen depending on the message length which is fixed by the user.

Furthermore, Wang *et al.* have experimented, using different trellises, the robustness of the embedding on synthetic signals, with a simple blind additive embedding. The trellises have from 1 to 64 states with 1 to 128 arcs per states. Wang *et al.* [9] show that the configuration that gives the best robustness, for a comparable number of codeword, is the one with the largest number of state and with a number of arcs per state lower than or equal to the number of state. A trellis with 64 states and 64 arcs per state is a choice that gives good robustness results but in counterpart the computational complexity during the embedding is relatively high (see Section II-C).

B. Informed coding

Informed coding has been introduced in watermarking community around 1998 [3], [4]. Previous non-informed approaches such as Spread-Spectrum [13] usually do not take the *host signal* **x** into account in order to choose the codeword $\mathbf{c}^*$. The nice property of *informed coding* approaches is that theoretically, with Gaussian assumptions, and a high dimension random codebook, the *host signal* **x** does not influence the channel capacity [2]. Thus, with those assumptions, the

only capacity limitation comes from the attack power. With a non-informed approach and those assumptions, the capacity is limited by the *host signal* power and the attack power.

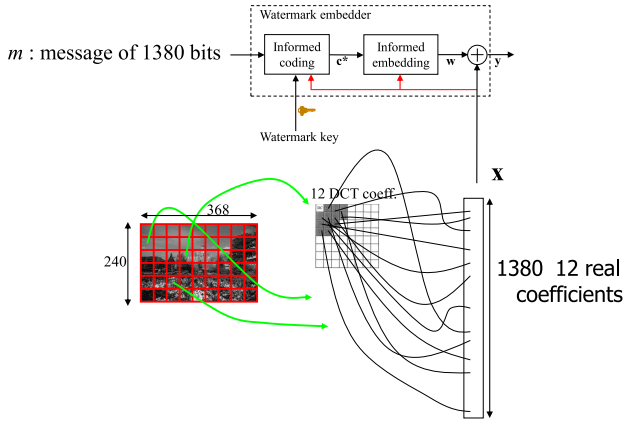


Fig. 5. Dirty Paper Trellis Codes apply on a 240×368 image.

The DTPC [1] belongs to the informed category. In the original scheme, an image is 8×8 DCT transformed, the twelve first ACs coefficients of each DCT blocks are extracted and pseudo-randomly ordered in a vector $\mathbf{x}$ of size $12 \times N/64 = 3 \times N/16$ (with N the image size). Fig. 5 illustrates the *host signal* $\mathbf{x}$ construction, the *informed coding* and the *informed embedding* steps on a 240×368 image.

The set of all the trellis paths i.e. all the possible outputs sequences, is the codebook $\mathcal{C}$ of the coder. A codeword $\mathbf{c}^i \in \mathcal{C}$ is the resultant coding of a message m. The *informed coding* is a way to choose the codeword $\mathbf{c}^i$ (encoding a given message m) the closest (for a given distance) to the *host signal* $\mathbf{x}$. Thus, *informed coding* allows to encode a message m by taking into account the *host signal* $\mathbf{x}$.

In the DPTC algorithm, given a message m, the *informed coding* is achieved:

- by pruning the trellis in order to keep the only valid paths. Thus, for a given transition, there are only the 0 input arcs or the 1 input arcs;
- by running a Viterbi decoder algorithm on this pruned trellis in order to find the closest codeword $\mathbf{c}^*$. The distance used in order to compare the codewords with the original host $\mathbf{x}$ is the scalar product. The Viterbi decoder thus retains the path (i.e. the codeword $\mathbf{c}^*$) of highest correlation with the *host signal* $\mathbf{x}$.

C. Informed embedding

In the original DPTC algorithm [1], a Monte Carlo approach is used in order to displace the *host signal* $\mathbf{x}$ into the Voronoï region of the codeword $\mathbf{c}^*$. This embedding is achieved in order to meet a given robustness. Moreover the modification of $\mathbf{x}$ is achieved by taking into account the psycho-visual degradation by using the Watson perceptual measure [14].

The monte-carlo principle is iterative and consists of attacking and counter-attacking a watermarked signal $\mathbf{y}$. The attack is achieved by gradually adding a Gaussian noise of increasing power, in order to do fail the decoder (bit errors on the message), and be under the threshold robustness value (see [1]). The counter attack is achieved by modifying the watermarked signal in order to resist to the previous successful attack. The iterative process is stopped after a sufficient number of successive unsuccessful attacks.

The Monte Carlo approach requires to run the Viterbi algorithm a high number of times. Even with the proposed optimizations in [1], the computational complexity is very high and this is at present a strong brake for intensive experiments studies[1]. The DPTC watermarking scheme is thus seriously competed with faster quantization-based approaches [5], [16].

Other sub-optimals approaches have been proposed in order to reduce the computational complexity. Wang *et al.* [9] uses an additive approach such that $\mathbf{y} = \mathbf{x} + \alpha\ \mathbf{c}^*$. This allows intensive experiments but a weak embedding capacity.

In [17], Wang *et al.* modifies the *host signal* $\mathbf{x}$ in order to displace it, in the Voronoï region of $\mathbf{c}^*$ but exactly in the same direction of $\mathbf{c}^*$ such that $\mathbf{y} = \frac{||\mathbf{x}||}{||\mathbf{c}^*||}\mathbf{c}^*$. With such an approach, and supposing that all the codewords own the same norm $||\mathbf{c}^*||$, the closest codeword found (at the decoder) in the watermarked signal $\mathbf{y}$ is $\mathbf{c}^*$. Indeed, the correlation of any codeword $\mathbf{c}^i$ is:

$$\begin{aligned} \forall \mathbf{c}^i \in \mathcal{C}, \mathbf{y}.\mathbf{c}_i &= ||\mathbf{y}||.||\mathbf{c}^i||.\cos\theta_i \\ &= ||\mathbf{x}||.||\mathbf{c}^*||.\cos\theta_i, \end{aligned}$$

with θ_i the angle between $\mathbf{y}$ and $\mathbf{c}^i$. The correlation $\mathbf{y}.\mathbf{c}^*$ is the highest one, since the angle is null. The Wang *et al.* approach [17] is interesting but does not take into account the degradation aspect of the *host signal*. The robustness is strong since the watermarked signal is exactly in the center of the Voronoï region but the modification of $\mathbf{x}$ is too high and inacceptable for real images.

A less degrading approach has been proposed by Lin *et al.* [18]. Fig. 6 illustrates Voronoï regions in case of an embedding space of size 3. Supposing that the codewords own the same norm, each black dot on the sphere represents a codeword. A Voronoï region of a codeword is a space area delimited by planes whose intersections with the sphere are drawn by edges surrounding the codewords.

In the Lin *et al.* [18] approach, once the closest codeword $\mathbf{c}^*$ has been computed (see section *informed coding* II-B), the closest codeword **c'** to $\mathbf{c}^*$ is computed. This is achieved by modifying the Viterbi algorithm in order to extract the second best path. The first most correlated codeword to $\mathbf{c}^*$ is $\mathbf{c}^*$ and the second one is **c'**.

Knowing $\mathbf{c}^*$ and **c'**, we deduce the circular hyper-cone whose apex is 0, whose axis is the vector $\mathbf{c}^*$, and whose surface goes through ($\mathbf{c}^*$+**c'**)/2 (see Fig. 6). The host vector $\mathbf{x}$ is then projected inside the cone onto the hyper-hyperboloid defined by a fixed robustness. See Cox, Miller and Bloom for more details [10].

[1]The experimentation (payload = 1/64 bpp, 100 embeddings, and 5000 attacks, 8-bits images of 256×256) takes more than one week running on a single-core PC with 3GHz. If we would set the SSIM [15] quality metric, for each image, it would take around 10 times longer. Moreover, in order to obtain satisfying results, the numbers of iterations have to be increased, and the granularity of the attack power has to be attenuated. It would result in a huge increase of time. The DPTC is clearly not suitable for payload of 1/64 bpp.

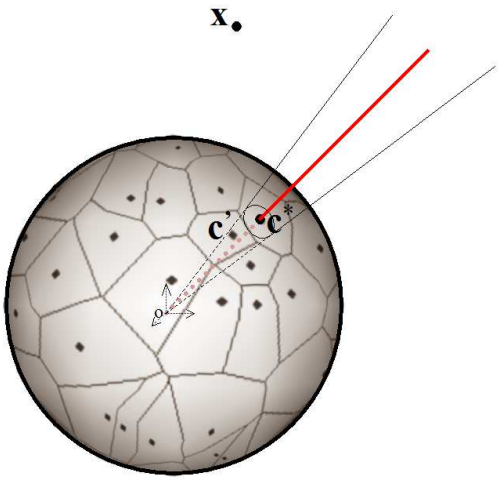


Fig. 6. 3D representation of Voronoï regions and Lin *et al.* [18] hyper-cone.

The Lin *et al.* approach [18] is a very clever way to reduce the computational complexity and ensure a tradeoff between robustness and distortion. Nevertheless, the watermarked signal $\mathbf{y}$ is often too far from the optimal robustness-distortion point. It follows a too strong distortion during the embedding process [19]. This may geometrically be explained on the Fig. 6 where a vast part of the Voronoï region is not used as embedding space whereas the optimum point could have been in this zone.

III. New embedding approach

In this section, we present our proposed method: the embedding space, the embedding approach and a psychovisual extension.

A. Embedding space

Fig. 7 illustrates our proposition : the Rotation-Based Dirty Trellis Codes (RB-DPTC). Our new embedding space is obtained by first, a wavelet transform of the image, and second, projections of the *host signal* $\mathbf{x}$ of dimension N_{wlt} ($\mathbf{x}$ is the concatenation of sub-bands coefficients except LL sub-band's coefficients[2]) onto N_{sec} carriers (noted $\mathbf{u}_i$ with $i \in [1, N_{sec}]$) in order to obtain the vector $\mathbf{v_x}$ of dimension N_{sec}. Carriers are normalized bipolar pseudo-random sequences. In high dimension, carriers are quasi-orthogonal. A projection is just a scalar product.

Note that during the extraction process, each projection brings together many wavelet coefficients into a single coefficient of the embedding space $\mathbf{v_x}$. The projections increase the Watermak-to-Content-Ratio (WCR) and thus improve the robustness of the scheme. The concepts of projections and retro-projections (spreading) come from the non-informed techniques of Spread Spectrum [13].

Also note that the complexity of the projection is reduced to a linear complexity with a Space Division Multiplexing approach [21] [3]. The obtained vector $\mathbf{v_x}$ of dimension N_{sec} may then be used for the *informed coding* (see Section II-B) and *informed embedding* (see Section II-C).

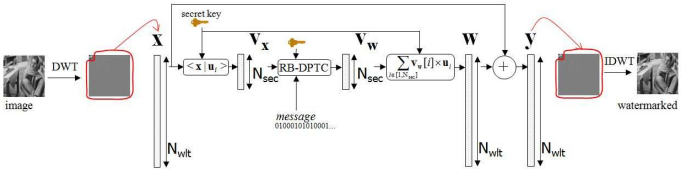


Fig. 7. Our Rotation-Based Dirty Paper Trellis Codes (RB-DPTC) scheme.

This embedding space allows to spread the watermark signal on almost all the frequency domain. Moreover, the projections onto N_{sec} carriers give to the embedding space a Gaussian aspect (Central Limit Theorem) which is known for its good property for the channel capacity[4] [2]. Finally, the wavelet domain is known for its good psycho-visual properties and introduces less disturbing effects than the block effects from the DPTC domain [1].

B. Embedding algorithm

The *informed coding* is the same as the original one (see section II-B) but is achieved with the host vector $\mathbf{v_x} \in \mathbb{R}^{N_{sec}}$ (secret space). After achieving the *informed coding*, the codeword $\mathbf{c}^* \in \mathbb{R}^{N_{sec}}$ is extracted. As explained in Section II-C, the solution proposed in Lin *et al.* [18], in order to speed-up the embedding and keep a good robustness-distortion tradeoff, is not satisfying since the degradation is too strong [19]. On the contrary, our approach gives a good compromise between complexity, robustness and distortion.

Remember that **at the decoder**, the most correlated codeword $\tilde{\mathbf{c}^*} \in \mathbb{R}^{N_{sec}}$ is obtained by running the Viterbi algorithm on the "unpruned" trellis. This codeword $\tilde{\mathbf{c}^*}$ belongs to the codebook $\mathcal{C}$ and maximizes the correlation with the attacked-watermarked vector $\tilde{\mathbf{v_y}} \in \mathbb{R}^{N_{sec}}$ such that:

$$\begin{aligned} \tilde{\mathbf{c}^*} &= \arg\max_{\mathbf{c}^i \ \forall \mathbf{c}^i \in \mathcal{C}} (\tilde{\mathbf{v_y}}.\mathbf{c}^i) \\ &= \arg\max_{\mathbf{c}^i \ \forall \mathbf{c}^i \in \mathcal{C}} (||\tilde{\mathbf{v_y}}||.||\mathbf{c}^i||.\cos\theta_i), \end{aligned} \quad (1)$$

with θ_i the angle between $\tilde{\mathbf{v_y}} \in \mathbb{R}^{N_{sec}}$ and $\mathbf{c}^i \in \mathbb{R}^{N_{sec}}$. Knowing that all the codewords own the same norm, the Viterbi algorithm extracts the codeword $\mathbf{c}^i$ owning the smallest angle θ_i with $\tilde{\mathbf{v_y}}$. A low-power Additive White Gaussian Noise attack is uncorrelated to $\mathbf{c}^*$ and thus does not modifies the initial angle θ_i. During the extraction, assuming that the attack was an AWGN, we should retrieve the codeword $\mathbf{c}^*$ used at the embedding.

In order to embed the message m, it is thus sufficient to reduce the angle between the host vector $\mathbf{v_x}$ and the codeword

[2]In many watermarking approaches such as BA [20], the coefficients of low frequencies are not used to avoid that the image degradation, due to the message embedding, be perceptible.

[3]The Space Division Multiplexing approach (SDM) [22] consists to apply a shuffling to the *host signal* $\mathbf{x}$, to divide the vector $\mathbf{x}$ into disjoint *regions* of quasi-equal sizes, and to use a carrier by *region*. The computational complexity is thus linear in function of the image size N instead of being quadratic.

[4]Costa [2] uses Gaussian hypothesis on the host source distribution in order to demonstrate that the source does not influence the channel capacity (the capacity is the quantity of bits that may be transmit without any errors).

$\mathbf{c}^*$ until obtaining the smallest angle regarding all the other angles $(\widehat{\mathbf{v_x}, \mathbf{c}^i})$.

In order to reduce the angle between $\mathbf{v_x} \in \mathbb{R}^{N_{sec}}$ and $\mathbf{c}^* \in \mathbb{R}^{N_{sec}}$, we first express these two vectors in the Miller, Cox and Bloom (MCB) plane [10]. Fig. 8 illustrates this MCB plane. The MCB plane is defined by an ortho-normalized basis ($\mathbf{v_1}$, $\mathbf{v_2}$), with $\mathbf{v_1} \in \mathbb{R}^{N_{sec}}$ and $\mathbf{v_2} \in \mathbb{R}^{N_{sec}}$, such that $\mathbf{v_x}$ and $\mathbf{c}^*$ belong to that plane (Gram-Schmidt algorithm):

$$\begin{aligned} \mathbf{v_1} &= \frac{\mathbf{c}^*}{||\mathbf{c}^*||}, \\ \mathbf{v_2} &= \frac{\mathbf{v_x} - (\mathbf{v_x}.\mathbf{v_1})\mathbf{v_1}}{||\mathbf{v_x} - (\mathbf{v_x}.\mathbf{v_1})\mathbf{v_1}||}. \end{aligned}$$

In the MCB plane, the 2D coordinates of the host vector $\mathbf{v_x}$ are:

$$\begin{aligned} \mathbf{v_x^{2D}}(1) &= \mathbf{v_x}.\mathbf{v_1}, \\ \mathbf{v_x^{2D}}(2) &= \mathbf{v_x}.\mathbf{v_2}, \end{aligned}$$

and the 2D coordinates of the codeword $\mathbf{c}^*$ are:

$$\begin{aligned} \mathbf{c^*_{2D}}(1) &= ||\mathbf{c}^*||, \\ \mathbf{c^*_{2D}}(2) &= 0. \end{aligned}$$

A rotation of the host vector $\mathbf{v_x^{2D}} \in \mathbb{R}^2$ of a θ angle in the MCB plane is such that:

$$\begin{aligned} \mathbf{v_y^{2D}}(1) &= \cos\theta.\mathbf{v_x^{2D}}(1) - \sin\theta.\mathbf{v_x^{2D}}(2), \\ \mathbf{v_y^{2D}}(2) &= \sin\theta.\mathbf{v_x^{2D}}(1) + \cos\theta.\mathbf{v_x^{2D}}(2). \end{aligned} \quad (2)$$

The vector $\mathbf{v_y} \in \mathbb{R}^{N_{sec}}$ is then obtained by expressing $\mathbf{v_y^{2D}} \in \mathbb{R}^2$ in the N_{sec} dimension space:

$$\mathbf{v_y} = \mathbf{v_y^{2D}}(1).\mathbf{v_1} + \mathbf{v_y^{2D}}(2).\mathbf{v_2}$$

If we reduce the absolute angle between the host vector $\mathbf{v_x}$ and the codeword $\mathbf{c}^*$ in the MCB plane, it increases the correlation $\mathbf{v_x}.\mathbf{c}^*$. With a dichotomous approach on rotation angle, one can rapidly find a Voronoï frontier i.e the frontier angle θ_f. The algorithm obtaining this Voronoï frontier is iterative and dichotomous[5]

Vectors $\mathbf{v_x}$ and $\mathbf{c}^*$ are the inputs of the algorithm. Let us define two variables $\theta_{max} \leftarrow 0$ and $\theta_{min} \leftarrow (\widehat{\mathbf{v_x}, \mathbf{c}^*})$ (note that $\theta_{min} \leq 0$) and set variable $\theta_f \leftarrow (\theta_{min} + \theta_{max})/2$. The algorithm repeats sequentially step 1 to step 3 (there are less than 10 iterations):

1) rotate $\mathbf{v_x}$ (in the MCB plane) of an angle θ_f in order to obtain $\mathbf{v_y}$ (see equation 2),
2) run the Viterbi decoder with $\mathbf{v_y}$ as input. If the extracted message is error free, $\mathbf{v_y}$ belongs to the Voronoï region, otherwise it does not.
3) modify the rotation angle depending on Voronoï region's belonging: if the extracted message (at step 2) was error free then $\theta_{min} \leftarrow \theta_f$ else $\theta_{max} \leftarrow \theta_f$; Update the rotation angle $\theta_f \leftarrow (\theta_{min} + \theta_{max})/2$. Return to 1 while $|\theta_{min} - \theta_{max}|$ is greater than a given threshold.

[5]In computer science, a dichotomous approach (dichotomic search approach with a "divide and conquer" strategy) is an iterative or recursive search algorithm, where, at each step, we divide in two parts a research space which becomes restricted to one of these two parts. In our approach we are looking for angle θ_f in the research range $[(\widehat{\mathbf{v_x}, \mathbf{c}^*}), 0]$.

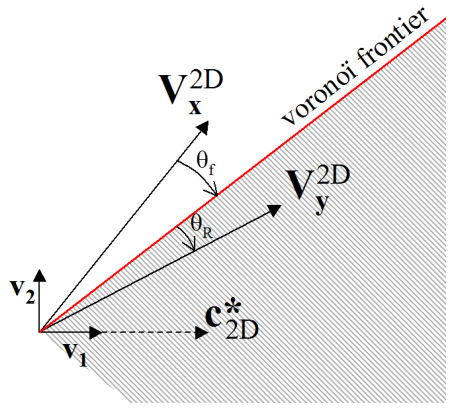


Fig. 8. Rotation-based embedding in the Miller, Cox and Bloom plane.

Once the frontier angle θ_f in the MCB is found, we improve the embedding robustness by penetrating inside the Voronoï region with a given angle θ_R. Our *informed embedding* is thus a rotation of the *host vector* $\mathbf{v_x}$ of an oriented angle equals to the $\max(\theta_f + \theta_R, (\widehat{\mathbf{v_x}, \mathbf{c}^*}))$. Fig. 8 illustrates $\mathbf{v_x}$, $\mathbf{v_y}$, θ_f and θ_R in the MCB plane. Note that a safety criteria proposed in [23] to counter the attack by principal component analysis of [24], and try to improve the algorithm of Broken Arrows [20] is to impose $||\mathbf{v_x}|| = ||\mathbf{v_y}||$. This is exactly what we do by making a rotation of $\mathbf{v_x}$; the norm of $\mathbf{v_y}$ equals the norm of $\mathbf{v_x}$.

We then compute the watermark vector $\mathbf{v_w} = \mathbf{v_y} - \mathbf{v_x}$, retro-project it onto carriers in order to obtain the watermark signal $\mathbf{w}$:

$$\mathbf{w} = \sum_{i=1}^{i=N_{sec}} \mathbf{v_w}(i).\mathbf{u}_i, \quad (3)$$

with $\mathbf{v_w}(i)$, the i^{th} component from vector $\mathbf{v_w}$, and $\mathbf{u}_i$ the i^{th} secret carrier defined in Section III-A. Then, we compute the watermarked signal $\mathbf{y} = \mathbf{x} + \mathbf{w}$. The inverse wavelet transform of $\mathbf{y}$ gives the watermarked image. At the extraction we project wavelet coefficients onto secret carriers and then retrieve the closest codeword (and thus the message) from the codebook $\mathcal{C}$ thanks to the Viterbi algorithm.

Fig. 9 shows the proposed embedding solution and the Lin *et al.* one [18] on a 2D Voronoï scheme. The Lin *et al.* embedding region is inside the circle centered on $\mathbf{c}^*$ codeword. As discussed previously, this embedding region is a strong reduction of the Voronoï region. With our approach, we displace the host vector $\mathbf{v_x}$ inside the Voronoï region, toward $\mathbf{c}^*$, and with a fixed angle penetration. The advantage of the approach on real data i.e. images, is that PSNR may be greater than 42 dB which is not the case with Lin *et al.* approach which gives an average maximum PSNR of 34 dB [19] on the 100 first images from the BOWS-2 database[6].

C. A psychovisual extension

In order that the impact of the watermarking is psycho-visually invisible, it is classical to "shape" the signal thanks

[6]The BOWS-2 database is downloadable at http://bows2.gipsa-lab.inpg.fr/.

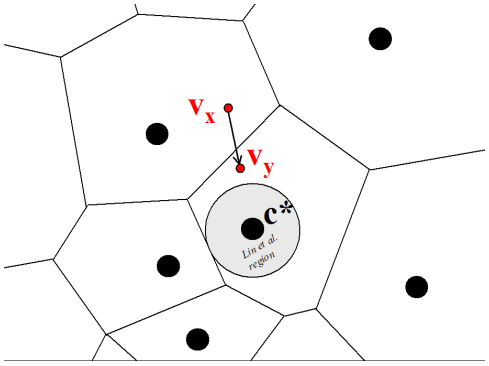
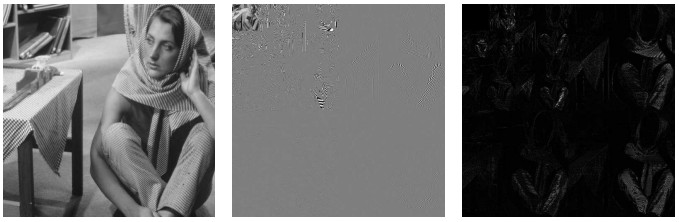


Fig. 9. 2D representation of embedding solutions.

to a psychovisual masking. Roughly speaking, the watermark signal strength should be reduced in uniform regions and should be increased in contours areas or textured regions. Our mask is obtained, first, by filtering the image with a high-pass filter, second, by applying a wavelet transform to the filtered image, and third, by linearly transforming the absolute values of the wavelet coefficients between $[1, 8]$. The mask for the Barbara 8-bit image (Figure 10.a), crop to a 512×512 image, is given in Figure 10.c.; values are multiplied by 255. The Figure 10.b gives the wavelets decomposition in 3 levels with the 9/7 Daubechies decomposition; Values are centered on the value 127. Note that this mask is recomputed at the extraction side.



(a) Barabara image (b) Wavelet decomposition (c) Mask

Fig. 10. Illustration of the psychovisual mask in the wavelet domain

The psychovisual extension briefly presented here is discussed more in detail in [25]. Note that in this psychovisual extension we also added an encoding of the message by an error correcting code of rate $1/2$.

Fig. 11 shows the general embedding scheme with the use of a psychovisual mask. Referring to that figure, there are three majors steps compared to the scheme of Fig. 7:

1) the construction of a psychovisual space $\mathbf{x_{psy}}$. In this space, coefficients are psychovisually equivalents. One can fairly embed in each coefficient. This psychovisual space is such that: $\forall i \in [1, N_{wlt}], \mathbf{x_{psy}}[i] = \mathbf{x}[i]/\alpha[i]$, with α the psychovisual mask;
2) the shaping of the watermark signal with the mask α: $\forall i \in [1, N_{wlt}], \mathbf{w}[i] = \mathbf{w_{psy}}[i] \times \alpha[i]$. This shaping reduces the psychovisual impact of watermarking in the areas where it would have been visible. For example, the value of α will be small in regions where light intensity is uniform to reduce the power of $\mathbf{w}$ in those areas;
3) the shaped watermark embedding. This embedding is such that: $\forall i \in [1, N_{wlt}], \mathbf{y}[i] = \mathbf{w}[i] + \mathbf{x}[i]$.

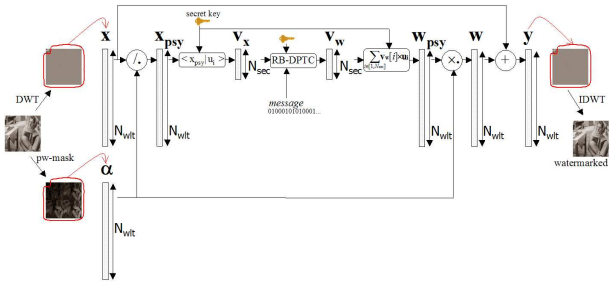


Fig. 11. Embedding scheme with a psychovisual mask.

In the DPTC [1] or BA (Broken Arrows) [20] algorithms, the psychovisual impact is taken into account during the embedding, and it is not necessary to recompute a mask at the decoder. The approaches used in BA and DPTC are nevertheless difficult to reapply in the rotation-based DPTC approach, either because the assumptions are not appropriates, either because the computational complexity is too high. In the [26] approach, the shaping of the watermark signal is achieved in the phase of retro-projection onto carriers. With our approach, we can use any psychovisual mask from the literature and moreover each coefficients from the embedding space are psychovisually equivalent.

The computation of the mask, during extraction, is a delicate phase. The mask must be the same as that used during embedding. If the masks are too different the message might be incorrectly extracted. Thus, the mask must also have properties of robustness to various attacks of a watermarking system. Other masks, in the wavelet domain, may been used like the mask proposed by Xie and Shen [27], which is an improvement of the well known pixel-wise masking model of Barni *et al.* [28], but some experiments show that the robustness is not improved [25]. There is still research to achieve in the future about robust psychovisual masks.

Thus, the decoder extracts the wavelet vector $\tilde{\mathbf{y}}$ from the watermarked-attacked image, divides each component i by $\alpha[i]$ (α is re-computed at the decoder), and projects the resulting vector onto the secret carriers in order to obtain a vector $\tilde{\mathbf{v_y}}$. All those steps are the same than those of embedding process (and are shown in Figure 11). As explained in Section III-B, the most correlated codeword $\tilde{\mathbf{c}}^* \in \mathbb{R}^{N_{sec}}$ is obtained by running the Viterbi algorithm on the "unpruned" trellis. When running the Viterbi algorithm, we are searching to the codeword $\tilde{\mathbf{c}}^*$ that belongs to the codebook $\mathcal{C}$ and that maximizes the correlation with the attacked-watermarked vector $\tilde{\mathbf{v_y}}$ (see Equation 1). The path in the trellis associated to the most correlated codeword $\tilde{\mathbf{c}}^*$ is the message $\tilde{m}$ that we retrieve.

D. Discussion on security aspects

The recent work of Bas and Doërr [8] about security of DPTC shows that in the Kerckhoffs's framework [29], i.e. when embedding and extracting algorithms are known by an

attacker, the *trellis codebook* may be retrieved[7] by observing a large number of watermarked images. Those conclusions are drawn based on a simplified version of the DPTC algorithm (non pseudo-random-ordering of DCT coefficients) but show a certain security weakness of DPTC [1]. The private space, that we use in this paper, allows to hide the structure of the trellis. A security attack based on the principle exposed in [8] is thus at least as difficult to lead with our proposition. Moreover, it is certainly very difficult to estimate the secret projections in the same way as [24] since there is a high number of codewords (with a trellis made of 128 states and 128 arcs per state and with a payload of 1024 bits, there are more than 10^{387} codewords). Note that there is a big difference between attacks on the robustness and attacks on the security. An attack on security consists to estimate the secret parameters (a secret key, the secret vectors, ...). In this article we will not address security attacks. This will be discussed in future work. For a better understanding of the difference between robustness and security, the reader will find practical scenarios on the website of the BOWS2 competition [30], and interesting definitions of security in [31], [32].

IV. Results

The experiments were performed on the first 100 images of the BOWS-2 database[8] with images resized to 256×256[9]. These images are grayscale photos taken by amateurs and coded on 8 bits.

Four types of attacks to robustness have been tested: the Gaussian noise attack, the filtering attack, the valumetric scaling attack, and the JPEG attack. In case of real images, the use of those four types of attacks in order to analyze the robustness is a classical methodology [1]. For the analysis of one image, those four types of attacks necessitate 50 different executions with varying parameters. The four attacks are described in detail in [1]. Note that we have also achieved an evaluation of a JPEG2000 lossy compression attack, with the JasPer software [33]. None of the three algorithms are able to face this attack even at very small compression rates. This is mainly because the embedding space, for the three algorithms, is not adapted. Moreover, we have chosen some difficult experimental conditions: a high payload (1/64 bpp) and a high SSIM[10] [15] (98%).

Let us remark that we do not address the malicious robustness attacks in the paper. A malicious attack on the robustness, like the Westfeld denoising attack [34], would suppress the watermark signal. Those attacks are borderline for high-rate watermarking schemes. The aim of the high-rate watermarking schemes is usually to propose a robust communication on a noisy channel and not on a malicious channel.

[7]More precisely, these are the coefficients attached to the arcs of the trellis that can be fairly well estimated.

[8]The BOWS-2 database is downloadable at http://bows2.gipsa-lab.inpg.fr/.

[9]The image sub-sampling has been achieved with the xnview program using Lanczos interpolation.

[10]SSIM is a classical measure well correlated to the Human Visual System. The SSIM values are real positive numbers lower or equal to 1. Stronger is the degradation and lower is the SSIM measure. A SSIM value of 1 means that the image is not degraded. To compute the SSIM value, we use the C++ implementation of Mehdi Rabah available at http://mehdi.rabah.free.fr/SSIM/.

The Bit Error Rate (BER) is computed from the extracted message and is equal to the number of erroneous bits divided by the total number of embedded bits. The BER is computed for each attack. We fixed the degradation to a SSIM [15] value of 98%. The payload is such that there is 1 bit embedded in 64 pixels such as the original DPTC algorithm [1]. The number of embedded bits is thus 1024 bits.

Three algorithms are competing: our psychovisual Rotation-Based algorithm (**PRB-DPTC**), multi-Hyper-Cube watermarking scheme (**MHC**) [35] which is a P-QIM like algorithm [36], and Turbo-TCQ algorithm (**T-TCQ**) [26]. For each algorithms, the payload and the SSIM value are the same. All those algorithms are multi-bit high rate watermarking schemes, and take into account the psychovisual impact of the watermark. Those algorithms have been defined and tested for real images, and not only on pure Gaussian signals. Moreover, they have a small $\mathcal{O}(size)$ complexity with $size$ the size of the image. The embedding computational time is around few seconds for a CIF 360×288 on a low cost laptop (Processor = Intel(R) Core(TM)2 Duo CPU P86000 2.4 GHz, RAM = 4GB).

For PRB-DPTC algorithm, the trellis structure owns 128 states with 128 arcs per state. Outputs arcs labels are drawn from a Gaussian distribution and there are 10 coefficients per output arc. Wavelet transform is a $9/7$ Daubechies with $l = 3$ decomposition levels. Except the LL sub-band, all the other sub-bands are used to form the *host signal* $\mathbf{x}$. With 256×256 images, the wavelet space size is thus $N_{wlt} = 64\ 512$ coefficients. In order to embed the 2×1024 bits (the correcting code rate is 1/2), with a trellis with $N_{arc} = 10$ coefficients per arcs, the private space size should be $N_{sec} = 2 \times 1024 \times 10 = 20\ 480$ coefficients. The inside angle penetration is tune in order to reach a SSIM = 98%. The selected psychovisual mask is the one based on a high-pass filtering [25].

The multi-Hyper-Cube watermarking scheme (**MHC**) [35], [36] is achieved in the DCT domain and is a TCQ-based watermarking approach using the Watson perceptual metric. The Turbo-TCQ (**T-TCQ**) [26] approach is also achieved in the DCT domain with a TCQ-based approach and the use of turbo principle coming from correcting codes domain.

A. Computational cost

In order to give an idea of the computational complexity we measured the CPU cost, averaged on 100 images, for an embedding in an image with a SSIM = 98% (this requires around 10 dichotomous iterations by image) for the three algorithms. The computer is a low cost laptop with a processor Intel(R) Core(TM) 2 Duo CPU at 2,4GHz with 4GB RAM. The results are given in the Table I. The MHC [35] approach is the faster with a cost below one second. This is a classical result for quantized based approaches. The T-TCQ [26] is around 5 times longer since it necessitates applying the turbo principle which is expensive. The RB-DPTC is around 12 times longer than MHC. Remark that the original approach [1] would take around 10 hours on the same PC with the same conditions. Our approach greatly reduces the computation cost.

Algorithm	CPU cost (seconds)
MHC [35]	0.98
T-TCQ [26]	5.73
PRB-DPTC	12.74

TABLE I
COMPARISON OF CPU COSTS FOR THE EMBEDDING IN AN 8-BITS IMAGE 256×256 WITH A SSIM = 98%.

B. Valumetric scaling attack

The valumetric attack modifies the pixels intensity of a scaling factor $\nu \in \mathbb{R}_+$. An intensity value v is transformed in $\nu.v$. Thus, for a scaling factor below 1 the image is darkened and for a scaling factor above 1 the image is brightened.

The results for the valumetric attack are given in Fig. 12. Usually, the T-TCQ [26] outperforms the other approaches, but for the valumetric one, it has very poor performances. This was already observed in [26] and it is a classical observation for quantization-based approaches. Indeed, the valumetric attack introduces desynchronizations: the quantized values are different from those computed at the embedding. In order to suppress this sensitivity the RDM trick [37] is used in MHC [35]. The PRB-DPTC has very good performance facing valumetric attack, especially for downscaling. This very good behavior was already observed in [1] and is still true with our rotation based approach. This comes from the low sensitivity to valumetric downscaling of the correlation measure used for the decoding in the Viterbi algorithm.

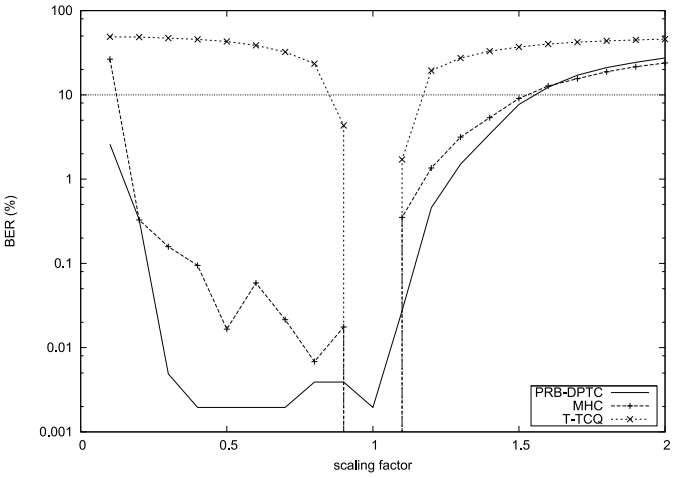


Fig. 12. BER for the valumetric scaling attack.

C. Low pass filtering attack

Fig. 13 shows BER results of a low pass filtering. The filter is a Gaussian filter of kernel size 9×9, with a 0 mean, a standard deviation σ and whose kernel coefficients are:

$$k(u,v) = \frac{1}{\sigma\sqrt{2\pi}} \times e^{\frac{-(u^2+v^2)}{2\sigma^2}}, \quad (4)$$

with u and v the line and column positions related to the center of the kernel.

The Turbo-TCQ [26] has very good performances since it is robust to a filtering below 0.5 standard deviation. MHC is a little bit less efficient but is robust to a power lower to 0.4. The PRB-DPTC owns very low BER under a power attack of 0.4. We should note that a more efficient correcting code would probably nullify the BER. We may conclude that the PRB-DPTC owns a similar behavior to MHC, for the filtering attack. Remark that low-pass filtering is an attack which destroys the high frequencies. The Turbo-TCQ, and in to a lesser extend MHC, are less sensitive because their embedding spaces, mainly use low frequency coefficients. The Turbo-TCQ generally outperforms other approaches due to the use of a turbo principle coming from correcting codes and ensuring near-optimal performances. Remark that Turbo-TCQ results are given in order to have a kind of upper bound for almost all the experiments (except for the valumetric downscaling attack).

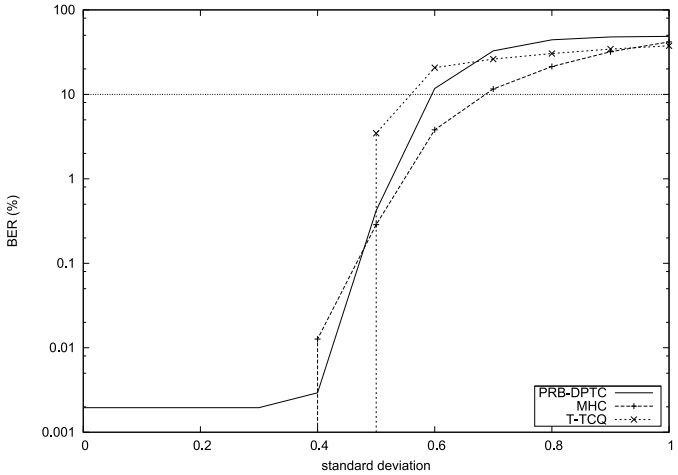


Fig. 13. BER for the Gaussian filtering attack.

D. Gaussian noise attack

The results for the Gaussian noise attack (zero mean and a standard deviation ranging from 0 to 10) are given in Fig. 14. Similarly to the filtering results, the MHC and the PRB-DPTC get the same performances. Those performances are similar because the quantization (principle in MHC), and the correlation (principle in PRB-DPTC), own a similar sensitivity to the Gaussian noise attack. Note that the obtained results are coherent to those obtained with synthetic signals (Gaussian). The T-TCQ obtains even better performances due to the turbo coding. We should point out that the turbo principle could be included in the PRB-DPTC in order to fill the performance gap with T-TCQ.

E. Jpeg attack

In order to simulate (in a reproducible way) a JPEG compression, we decompose the image in 8×8 DCT blocks. Each coefficient $c(i,j), i \in [0,7]$ and $j \in [0,7]$ of a DCT block is quantized and de-quantized such that:

$$c_q(i,j) = Q.q(i,j) \left\lfloor \frac{c(i,j)}{Q.q(i,j)} + 0.5 \right\rfloor, \quad (5)$$