TCQ PRACTICAL EVALUATION IN THE HYPER-CUBE WATERMARKING FRAMEWORK

Marc CHAUMONT[1,2] *and Dalila GOUDIA*[2]

[1] University of Nîmes, Place Gabriel Péri, 30000 Nîmes, France
[2] Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II
161, rue Ada, 34392 Montpellier cedex 05, France
contact: marc.chaumont@lirmm.fr

ABSTRACT

The Hyper-Cube watermarking has shown a high potential for high-rate robust watermarking. In this paper, we carry on the study and the evaluation of this quantization-based approach. We especially focus on the use of a Trellis Coded Quantization (TCQ) and its impact on the Hyper-Cube performances. First, we recall the TCQ functioning principle and we propose adapted quantizers. Second, we analyze the integration of the TCQ module in a cascade of two coders (resp. two decoders). Finally, we experimentally compare the proposed approach with the state-of-the-art of high-rate watermarking schemes. The obtained results show that our Multi-Hyper-Cube scheme always provides good average performances.

***Index Terms*—** High-rate robust watermarking scheme, Trellis Coded Quantization and watermarking joint scheme, Perceptual watermarking, Valumetric attack robustness.

1. INTRODUCTION

One of the most effective image quantization-based watermarking is currently the Hyper-Cube [1] scheme which is a derivation of the P-QIM [2] scheme (Perceptual-QIM). The embedding is achieved with a QIM [3] quantization-based approach. The RDM [4] principle is used in order to make the scheme less sensitive to the valumetric attack. This is achieved using a modified Watson model [5]. The Watson model also allows to take into account the psycho-visual impact due to embedding degradation[1].

Our paper is an extension of previous work on the Hyper-Cube [1] scheme. We propose to fill the gap between the trellis watermarking approaches [6, 7], and the quantization-based watermarking approaches [3, 8, 2] using a well designed TCQ module which replaces the current QIM module. Barni *et al.*[9] have proposed a close approach thanks to the use of a trellis and the RDM principle. The approach relies on a unique trellis, a vectorial quantization and a suboptimal research of the best path in the trellis. The experiments are achieved on i.i.d Gaussian signals and the results show an improvement compared to the RDM for a WNR (watermark noise ratio) close to 10 dB. Note that the solution has not been evaluated for real images, the RDM function is a L2 norm and thus it does not take into account the psycho-visual impact, and the approach gives less interesting results for WNR close to 0 dB (some future improvements are evoked in order to treat the problem).

In section 2, we recall the general principles of insertion and extraction. In section 3, we present the TCQ and watermarking joint approach. Finally, in Section 4 we present the results, and then we conclude.

2. THE HYPER-CUBE WATERMARKING FRAMEWORK

The Hyper-Cube [1] framework is summarized in Figure 1. The image is divided into 8x8 blocks, and one bit is embedded in each block. A DCT transform is applied on the current block $\mathbf{X}$, then the first n ACs coefficients from the zig-zag scan are stored in a vector called the host signal and noted $\mathbf{x}$. Next, the n coefficients from $\mathbf{x}$ are watermarked using scalar QIM. The n bits coming from the coded *message* $\mathbf{m}$ are thus embedded into $\mathbf{x}$. For each of the n coefficients of $\mathbf{x}$, the quantization step noted $\mathbf{\Delta}_i, i \in \{1, \ldots, n\}$, necessary for the watermarking, is a function of the modified Watson *slack* computed on a previously watermarked block.

The modified Watson *slack* associated with a DCT coefficient x in position $i \in \{0, ..., 63\}$ is [2]:

$$s(x,i) = max(t_L^M[i], |x|^{0.7} t_L^M[i]^{0.3}),$$

with t_L^M the brightness mask:

$$t_L^M[i] = t[i] \left(\frac{C[0]}{C_0}\right)^{0.649} \left(\frac{C_0}{128}\right),$$

[1] Note that there is really few papers that have proposed a high-rate (≈ 1 bit in 64 pixels) watermarking scheme which are evaluated on real images, take into account the psycho-visual impact, treats the valumetric attack problem, uses an informed approach, and sometimes include correcting codes. The well known high-rate approaches for real images taking all those criteria into account are DPTC [6] and P-QIM [2].

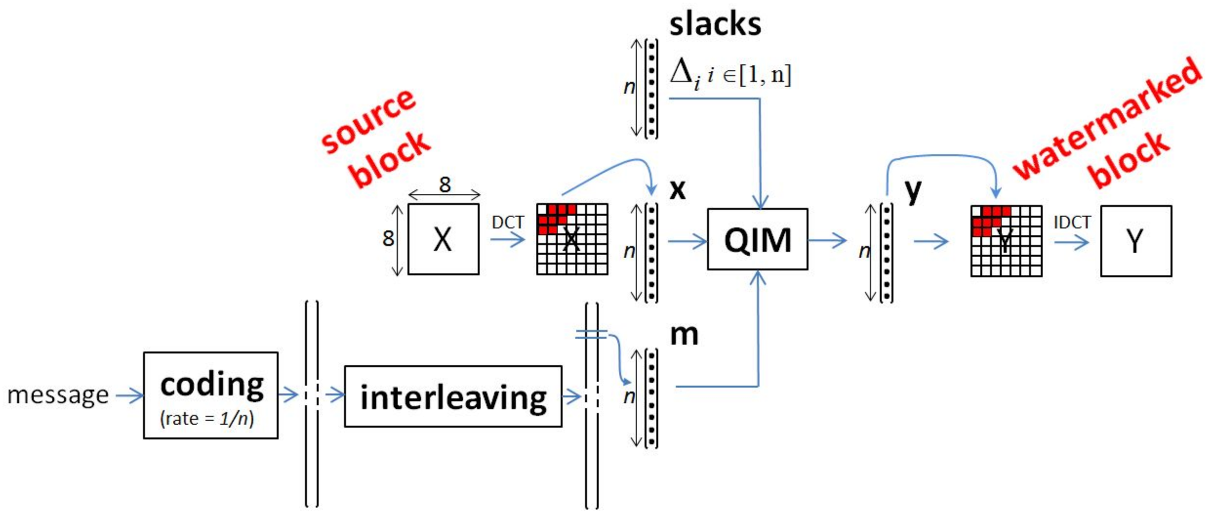

Fig. 1. Hyper-Cube general scheme for a 8x8 pixels block.

with $C[0]$ the DC coefficient of the DCT block, C_0 the average of all the DCs coefficients of the image, and $t[i]$ the sensitivity value with position i [5]. Compared to the Watson *slack*, the **modified** Watson *slack* linearly scales with coefficient scaling. A valumetric attack changing the amplitude of pixels with a scalar $\nu \in \mathbb{R}_+$ will thus scale the modified Watson *slacks* of a factor ν. This property allows a quantization-based watermarking system to be built, which is less sensitive to the valumetric attack.

The quantization step Δ_i used by the QIM module (see Figure1) in order to embed a bit $\mathbf{m}[i]$ in a coefficient $\mathbf{x}[i]$ is:

$$\Delta_i = G_{HC} \times s(x, i),$$

with $G_{HC} \in \mathbb{R}$ a constant tuning the embedding strength. Note that the *slack* $s(x, i)$ is computed on a previously watermaked block (the closest one between the upper or the left one [1]).

At the embedding, the binary message is encoded with a convolutional coder. The code rate of this convolutional coder is $1/n$ and it is represented by the "coding box" in Figure 1. Then, the resulting codeword is shuffled ("interleaving box" in Figure 1). The obtained vector is then split in small vectors of size n. Each vector of size n is hidden in a DCT 8x8 block. In Figure 1 and for the sake of simplicity, all the small vectors are noted $\mathbf{m}$. For a given DCT block, the watermarked signal $\mathbf{y}$ is obtained by quantifying each component of the host signal $\mathbf{x}$ with quantizers $\{Q_{\mathbf{m}[i]}\}_{i \in \{1,\ldots,n\}}$ such that[2]:

$$\forall i \in \{1, \ldots, n\}, \mathbf{y}[i] = Q_{\mathbf{m}[i]}(\mathbf{x}[i], \Delta_i),$$

with Δ_i the quantization step associated with the i^{th} coefficient and quantizers Q_0 and Q_1 defined such that:

$$\begin{aligned} Q_0(\mathbf{x}[i], \Delta_i) &= 2\Delta_i \times round\left(\frac{\mathbf{x}[i]}{2\Delta_i}\right), \\ Q_1(\mathbf{x}[i], \Delta_i) &= 2\Delta_i \times round\left(\frac{\mathbf{x}[i] - \Delta_i}{2\Delta_i}\right) + \Delta_i. \end{aligned}$$

At the extraction, there is a cascade of two decoders. The first decoder is fed with vectors $\mathbf{z}$ extracted from each watermarked-attacked DCT block. For each block, it calculates n Euclidean distances: the distances $d_0[i] = (\mathbf{z}[i] - Q_0(\mathbf{z}[i], \Delta_i))^2, i \in \{1, \ldots, n\}$ computed between the watermarked-attacked scalar $\mathbf{z}[i]$ and the scalar corresponding to an embedded bit 0, and the distances $d_1[i] = (\mathbf{z}[i] - Q_1(\mathbf{z}[i], \Delta_i))^2, i \in \{1, \ldots, n\}$ computed between the scalar $\mathbf{z}[i]$ and the scalar corresponding to an embedded bit 1. The second decoder is a convolutive decoder. It takes the distances from all the DCT blocks, de-interleaves the distances, and then carefully adds them in order to label the arcs of the trellis of the convolutional decoder. The decoding is then achieved using the Viterbi algorithm [10].

3. THE TCQ AND WATERMARKING JOINT SCHEME

3.1. The trellis

The aim of this paper is to evaluate the gain obtained by using a TCQ module replacing the QIM module (see Figure 1).

The TCQ (Trellis-Coded Quantization) is a quantization technique using a set of quantizers organized in a state machine and acting similarly to a convolutional coder. The state machine represents the possible transitions given an input symbol sequence. The state machine may be represented as

[2] Dithering may easily been included in those quantization functions. Theoretically, it should slightly increase the performances at low WNR.

it evolves in time with a trellis diagram. Usually, a trellis is constructed by placing all the states in column. Each transition is drawn with an arc between states at t time and states at $t+1$ time. By convention, the bold arcs represent a 1 input and the nonbold arcs a 0 input. An input coefficient causes a transition to a new state and outputs the result of the quantization of the input coefficient. Figure 2 shows a trellis diagram owning 4 states.

The transition function t of the trellis defines all the transitions such that:

$$\begin{array}{rcl} \mathcal{S} \times \{0,1\} & \longrightarrow & \mathcal{S} \\ t : (s, \mathbf{m}[i]) & \longmapsto & s', \end{array}$$

with $\mathcal{S} = \{0, 1, ..., S-1\}$ the set of states, $s \in \mathcal{S}$ the head of the transition arc, $s' \in \mathcal{S}$ the tail of the transition arc, and $\mathbf{m}[i], i \in \{1, \dots, n\}$, the i^{th} bit from $\mathbf{m}$.

Each arc is then labeled with a specific quantization function:

$$\begin{array}{rcl} \mathcal{S} \times \{0,1\} \times \mathbb{R} \times \mathbb{R} & \longrightarrow & U \\ Q : (s, \mathbf{m}[i], \mathbf{x}[i], \Delta_i) & \longmapsto & \mathbf{y}[i], \end{array}$$

with Δ_i the quantization step. For simplification, we will note the quantizers $Q_{\mathbf{m}[i]}(s, \mathbf{x}[i], \Delta_i)$. In Figure 2 each arc is labeled with a quantization function.

3.2. The quantizers definition

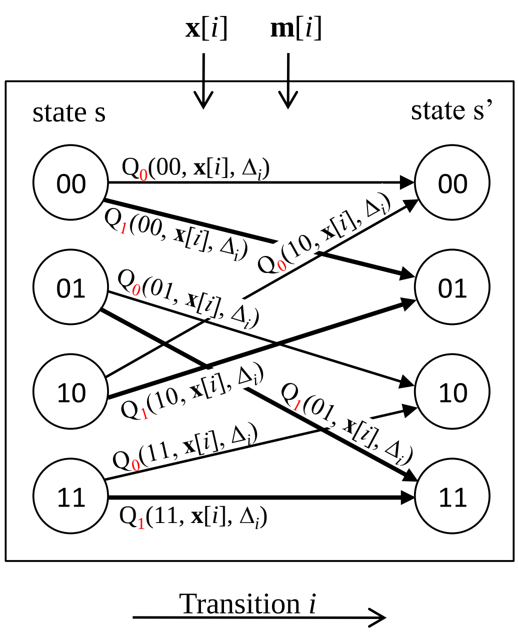

Fig. 2. The i^{th} transition step in a 4 states trellis.

The quantizers $Q_{m[i]}(s, \mathbf{x}[i], \Delta_i)$ are defined for a given state $s \in \mathcal{S}$, for an input scalar value $\mathbf{x}[i]$, for a quantization step Δ_i, and for an input bit $\mathbf{m}[i]$ equals to 0 or 1 by:

$$\begin{array}{rcl} Q_0(\mathbf{x}[i], s, \Delta_i) & = & 2\Delta_i \times round\left(\dfrac{\mathbf{x}[i]-\delta}{2\Delta_i}\right) + \delta, \\ Q_1(\mathbf{x}[i], s, \Delta_i) & = & 2\Delta_i \times round\left(\dfrac{\mathbf{x}[i]-\Delta_i-\delta}{2\Delta_i}\right) + \Delta_i + \delta, \\ with\ \delta & = & \dfrac{\Delta_i \times s}{S}. \end{array} \quad (1)$$

Figure 3 shows the partition of Real axis in the case of a four states trellis. Red circles represent codewords for an input bit 0 and red squares represent codewords for an input bit 1. For a given state $s \in \mathcal{S}$, the distance between codewords generated by a 0 transition and codewords generated by a 1 transition is equal to Δ_i. We can also remark that codeword are slightly translated between each states. This translation is due to the δ term in Equation 1. This particular setting makes the TCQ approach very interesting since depending on the path in the trellis, the quantization is not the same. It is increasing the probability to find a codeword close to the host value. At the end of the TCQ encoding, for a given robustness, the distortion is then lower than with a simple QIM approach. This translation term is very important for a functional watermarking system using a TCQ-watermarking joint system.

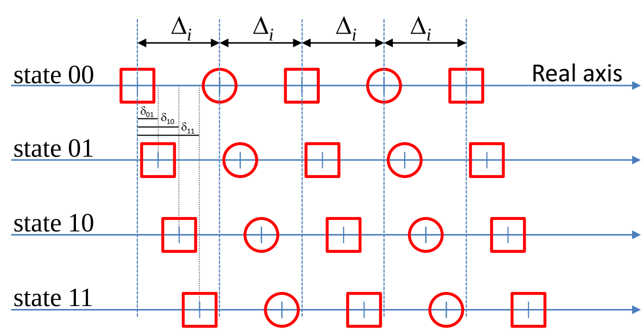

Fig. 3. Lattice illustration for a 4 states trellis. Red circles represent codewords obtained using quantizer Q_0 (Equation 1) and red squares represent codewords obtained using quantizer Q_1 (Equation 1).

3.3. The TCQ Decoding

The TCQ decoding is achieved with the Viterbi algorithm [10]. Let us define a sequence $\mathbf{y}$ composed of n bits which embed a message $\mathbf{m}$. Suppose that this sequence is degraded (attacked) by an Additive White Gaussian Noise (AWGN). The decoder receives a sequence $\mathbf{z}$ of n bits. The Viterbi decoder estimate from this sequence $\mathbf{z}$ the embedded bits $\mathbf{m}'[i], i \in \{1, \dots, n\}$ by maximizing the a posteri probability that a sequence was used for embedding. In practice, the

Viterbi algorithm finds the shortest path in the trellis associated to the TCQ code. This path corresponds to the message $\mathbf{m}'[i], i \in \{1, \ldots, n\}$.

For a perfect integration in the Hyper-Cube framework, the TCQ decoder should return soft information in order to feed the second decoder. The Viterbi decoder does not return soft information but a binary sequence. On the contrary, the BCJR [11] (also known as Maximum a Posteriori (MAP) algorithm or as forward-backward algorithm) allows to recover a soft information by computing the probability that an embedded bit was a 0 or a 1 for each transition. Initially BCJR suffered from strong complexity but this have been reduced with Soft-Output Viterbi Algorithm (SOVA) in 1989 [12], the log-MAP algorithm in 1995 [13]... One can thus use a BCJR (or a successor) in order to decode the TCQ codewords and then use this soft information to feed the second decoder. Unfortunately, experimental results give similar performances compared to the use of a cascade of two Viterbi decoders. We can also remark that a turbo approach similar to the one proposed in [14] is not possible since the same symbols may not been interleaved between different DCT block.

4. RESULTS

The experiments were performed on the first 100 images of the BOWS-2 database[3] with images resized to 256×256.[4] These images are grayscale photos taken by amateurs and coded on 8 bits.

Four attacks to robustness have been tested: the Gaussian noise attack, the filtering attack, the valumetric scaling attack, and the JPEG compression attack. The four attacks are described in detail in [6]. 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%[5]. The payload is fixed to 1 bit embedded in 64 pixels.

The two main families for **robust** multi-bit watermarking are the *lattice* codes also known as quantization-based codes and the *Dirty Paper Trellis Codes*. In order to analyze the performance of our Multi-Hyper-Cube watermarking scheme we use approaches best representing those two families. The *Dirty Paper Trellis Codes* [6] is represented by the PR-RB-DPTC [7] which has a small computational embedding complexity. The quantization-based approach is represented by the Hyper-Cube [1]. Finally, we also test the Turbo-TCQ [14] approach which is a mix between the two families (the *dirty paper trellis codes* and the *lattice code*) with the use of the turbo principle coming from correcting codes domain.

Note that those four algorithms are usable and realist techniques which 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 computational time is around few seconds for a CIF 360x288 on a low cost laptop.

The results for the valumetric attack are given in Figure 4. For all the other attacks, the Turbo-TCQ [14] outperforms the other approaches, but for the valumetric one, it has very poor performances. This was already observed in [14] and it is a classical observation for quantization-based approaches. In order to suppress this sensitivity we should use the RDM trick [4]. We could thus observe a better behavior for the Hyper-Cube framework [1] since the RDM principle has been integrated. For example, for a downscaling of a 0.9 factor, there is in average 0.018 bits erroneous on 100 transmitted bits for the Multi-Hyper-Cube whereas there is 4.33 bits erroneous on 100 transmitted bits. Note that globally Hyper-Cube [1] and Multi-Hyper-Cube (Multi-Hyper-Cube is the name of our proposition: Hyper-Cube + TCQ) curves are close but the Multi-Hyper-Cube has a null BER when there is no attack; This is not the case for the Hyper-Cube curves. Finally, note that the PR-RB-DPTC [7] has the best performance facing valumetric attack, especially for downscaling. This very good behavior was already observed in [6].

The results for the JPEG compression attack are given in Figure 5. Usually, the curves for Hyper-Cube and PR-RB-DPTC are often very close except for the JPEG compression attack where the PR-RB-DPTC is not enough robust. The original algorithm DPTC [6] is more robust but its complexity make it unpractical for such payload (1 bit embedded in 64 pixels). Moreover, other proposed improvements such that [16] are not really efficient in practice (see [17]). This shows that in practice, the Hyper-Cube is more interesting than the PR-RB-DPTC for high payload.

The results for the Gaussian noise attack are given in Figure 6 and for the filtering attack are given in Figure 7. Except the Turbo-TCQ [14] which has very good performances, the other approaches own similar performances.

To sum up, the two approaches which own good performances whatever the attack are the Hyper-Cube and the Multi-Hyper-Cube. The Multi-Hyper-Cube significantly improves the Hyper-Cube when there is an attack of very small power. Indeed, for the all 100 images, all the bits have been recovered for small power attacks. This result is interesting but we may not conclude that the Multi-Hyper-Cube algorithm outperforms the Hyper-Cube one. Indeed, when the power of all the attacks are increasing, the BER of the Multi-Hyper-Cube is not always lower than the BER of the Hyper-Cube. The TCQ approach allows quantizers to be addes, but in counter part, it probably adds more instability at

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

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

[5]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 decoding step when there is an attack of middle power.

The Multi-Hyper-Cube sometimes has behaviors which are similar to a correcting code. When the attack power is too strong the error probability is rapidly greater than 0,1 bit erroneous on 100 transmitted bits. This rapid growing of the BER is even more visible with the Turbo-TCQ [14] approach for the filtering and valumetric attack; the BER is suddenly growing to values greater than 1 bit erroneous on 100 transmitted bits. It is a classic behavior with approaches using near optimal correcting codes which are close to information theory bound. In conclusion, the Multi-Hyper-Cube gives a null BER for a small power attack but it does not outperform Hyper-Cube for attacks of middle power.

The Hyper-Cube framework and Multi-Hyper-Cube algorithm may still been improved. Remember that those two schemes has satisfying behavior for the four attacks and that it is not the case for the other ones. For example, the coefficients selected for the embedding may be more carefully studied and selected. A clever collaboration of the two decoders in the cascade of decoding may improve performances. Finally, a vectorial QIM or a spreading may probably slightly increase the performances.

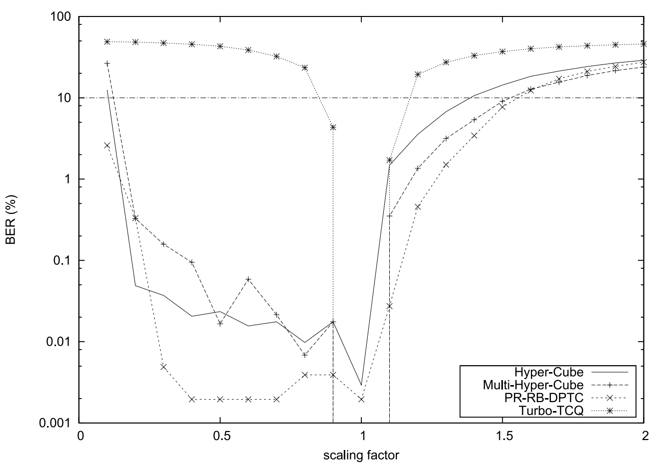

Fig. 4. BER for the valumetric scaling attack.

5. CONCLUSION

This paper presents the study of the integration of a TCQ module inside the Hyper-Cube watermarking framework. The QIM module is replaced by a TCQ module. The TCQ acts as if the number of quantizers were increased. This allows the robustness to be increased for a fixed degradation. The obtained Multi-Hyper-Cube algorithm is compared to the state-of-the-art of high-rate robust watermarking schemes. The results show that the scheme reacts equally well to the Gaussian, filtering, JPEG compression, and valumetric attacks. This behavior is neither observed with the Dirty Paper Trellis Codes PR-RB-DPTC [7], which is very sensitive to

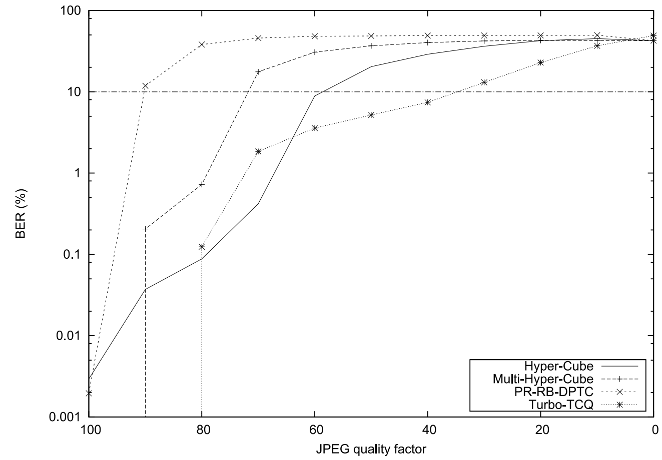

Fig. 5. BER for the JPEG compression attack.

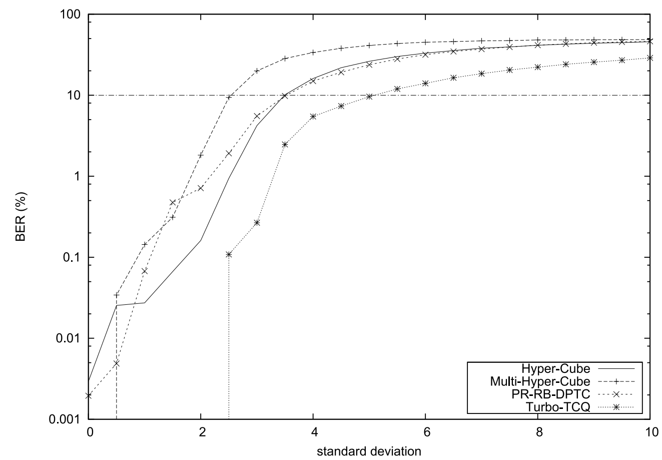

Fig. 6. BER for the Gaussian noise attack.

JPEG compression, nor with the Turbo-TCQ [14], which is very sensitive to valumetric attack. Moreover, for small power attacks, the BER of the Multi-Hyper-Cube is null. In the future we will deal with the selection of coefficients for the embedding. We also think that the vectorial QIM or the spreading approach may slightly increase the performances. Finally, a cleverer management of the two coders/decoders may increase the global performances.

6. REFERENCES

[1] M. Chaumont, D. Goudia, and W. Puech, “Hyper-Cube Watermarking Scheme,” in *Visual Information Processing and Communication II, Part of IS&T/SPIE 23th Annual Symposium on Electronic Imaging, VIPC2011, SPIE2011*, San Francisco, California, USA, Jan. 2011, vol. 7882, pp. 10–18.

[2] Q. Li and I. J. Cox, “Using Perceptual Models to Improve Fidelity and Provide Resistance to Valumetric

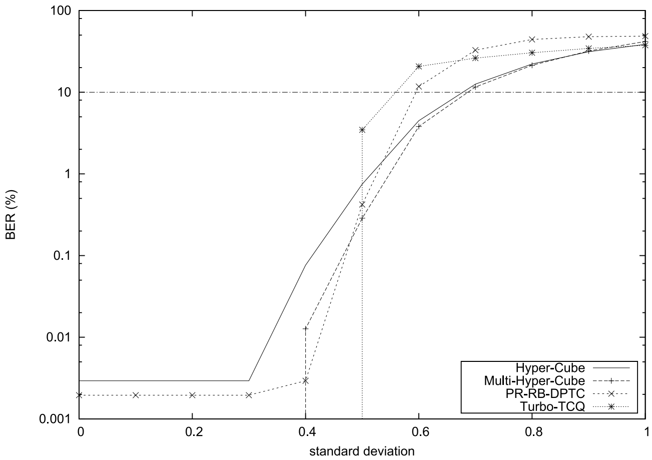

Fig. 7. BER for the Gaussian filtering attack.

Scaling for Quantization Index Modulation Watermarking," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 2, pp. 127–139, 2007.

[3] B. Chen and G. Wornell, "Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Infromation Embedding," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[4] F. Pérez-González, M. Barni, A. Abrardo, and C. Mosquera, "Rational Dither Modulation: A Novel Data-hiding Method Robust to Valuemetric Scaling Attacks," in *IEEE International Workshop on Multimedia Signal Processing, IWMSP'2004*, Sept. 2004, pp. 139–142.

[5] A. B. Watson, "DCT Quantization Matrices Optimized for Individual Images," in *Human Vision, Visual Processing, and Digital Display IV, SPIE'1993*, 1993, vol. 1913, pp. 202–216.

[6] M. L. Miller, G. Doërr, and I. J. Cox, "Applying Informed Coding and Informed Embedding to Design a Robust, High Capacity Watermark," *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 792–807, 2004.

[7] M. Chaumont, "Psychovisual Rotation-based DPTC Watermarking Scheme," in *17th European Signal Processing Conference, EUSIPCO'2009*, Glasgow, Scotland, Aug. 2009.

[8] J. J. Eggers, R. Bäuml, R. Tzschoppe, and B. Girod, "Scalar Costa Scheme for Information Embedding," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1003–1019, 2003.

[9] A. Abrardo, M. Barni, F. Perez-Gonzalez, and C. Mosquera, "Improving the performance of RDM watermarking by means of trellis coded quantisation," *IET Information Security, IEE Proceedings*, vol. 153, no. 3, pp. 107 –114, Sept. 2006.

[10] Andrew J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*, Addison-Wesley Wireless Communications, 1995.

[11] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[12] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *IEEE Global Telecommunications Conference. Communications Technology for the 1990s and Beyond. GLOBECOM'89*, Dallas, TX , USA, Nov. 1989, vol. 3, pp. 1680–1686.

[13] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *IEEE International Conference on Communications, ICC'95 , 'Gateway to Globalization'*, June 1995, vol. 2, pp. 1009–1013.

[14] G. Le Guelvouit, "Tatouage Robuste d'Images par Turbo TCQ," *Traitement du Signal*, vol. 25, no. 6, apr. 2009, **Source downloadable at http://www.gleguelv.org/wt/ttcq/**.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[16] L. Lin, I. J. Cox, G. Doërr, and M. L. Miller, "An Efficient Algorithm for Informed Embedding of Dirty Paper Trellis Codes for Watermarking," in *IEEE International Conference on Image Processing, ICIP'2005*, Genova, Italy, Sept. 2005, vol. 1, pp. 697–700.

[17] M. Chaumont, "A Novel Embedding Technique For Dirty Paper Trellis Watermarking," in *Visual Information Processing and Communication, Part of IS&T/SPIE 22th Annual Symposium on Electronic Imaging, VIPC'2010, SPIE'2010*, San Jose, California, USA, Jan. 2010, vol. 7543, pp. 38–46.