Lattice vector quantization with dead zone for an efficient combined compression and watermarking algorithm

L. Guillemot, T. Voinson, JM. Moureaux
CRAN - CNRS - Université Henri Poincaré, Nancy 1 -
Faculté des Sciences et Technologies - BP 239 - 54506 Vandoeuvre-lès-Nancy Cedex, France
ludovic.guillemot@cran.uhp-nancy.fr, voinson@iutsd.uhp-nancy.fr, moureaux@cran.uhp-nancy.fr

ABSTRACT

We propose here a new combined compression and watermarking technique. It is based on a Lattice Vector Quantizer with dead zone for a good robustness to attacks like white noise, as well as median filtering or compression.

1 Introduction

The development of the Information Society makes easy the exchange of multimedia data on the networks. In return, if digital data offer many advantages for transmission and storage purposes, they can be easily manipulated or corrupted with different bad intentions, like for example illegal duplications or unauthorized use of their content. In this context, the protection of digital data integrity has received special attention in the ten past years and many efforts have been done in the design of efficient digital watermarking techniques[5].

These techniques have to deal with two main constraints: invisibilty of the mark and robustness to piracy attacks or/and necessary post-processing. The trade-off between these constraints is not easy to tune. In particular, robustness to attacks is a difficult problem as several types of attacks exist. Among them, it is worth to notice that compression is an essential process in the data transmission or storage chain, and thus, one of the major attacks to watermarking. It can strongly damage the embedded mark because it tends to remove useless or non visible information whereas watermarking aims at embedding invisible information. In this paper, we use this contradiction to design a new combined compression and watermarking technique able to improve robustness against attacks like compression, as well as median filtering or white noise. The compression algorithm is based on discrete wavelet transform (DWT) associated to lattice vector quantization (LVQ) with dead zone (DZLVQ). This last point is of importance in our method, since the dead zone leads to improvement of both visual quality and robustness of watermarking. Figure 1 represents the global scheme of the proposed method. Watermarking is performed by substitution: the lattice vectors which are selected by the secret key are modified (marked) to be substituted to the initial vectors.

The paper is organized as follows. Section 2 gives details about LVQ with dead zone. In section 3, we focus on the watermarking algorithm and its robustness. Finally, simulation results and conclusion are given in section 4.

2 Lattice vector quantization with dead zone

2.1 LVQ definition and design

The main advantage of LVQ is its low complexity since there is no need to construct or to store the code book [6]. Indeed, a lattice in the R^n space is a set of regularly spaced vectors, whose coordinates are obtained by linear combinations of integers. Optimal lattices (in granular distortion sense) have been designed for several dimensiosn of vectors [2]. But the simplest and most common lattice is Z^n(which is non optimal).

As any compression application requires a finite bit rate, LVQ code books must be finite too. The lattice is thus truncated with an appropriate shape with respect to the source statistics. This permits to minimize the overload distortion due to the source scaling in the finite code book. It can be easily shown that a pyramidal truncation will be well suited to Laplacian data, whereas a spherical one will better suit Gaussian data. As in image coding, wavelet coefficients are often assumed to be i.i.d. random variables whose distribution is a Laplacian, we work here on pyramidal code books [4][1]. Note that our method is available for other distributions.

The quantization process is very fast, since in the case of Z^n, it only consists in computing the nearest integer of each component of the scaled source vectors. Finally the quantized vectors must be indexed and their index are variable length encoded. Indexing is a crucial process for compression applications. It has received special attention for years as it was a difficult problem because of the large size of the LVQ code books. Efficient solutions have been proposed for Laplacian or Gaussian distributions for example in [4][9].

Let $\mathbf{x}$ be a source vector. LVQ of $\mathbf{x}$ in a truncated lattice is given by: $(I \circ Q)(\frac{x}{\gamma})$, where γ is the scaling factor,

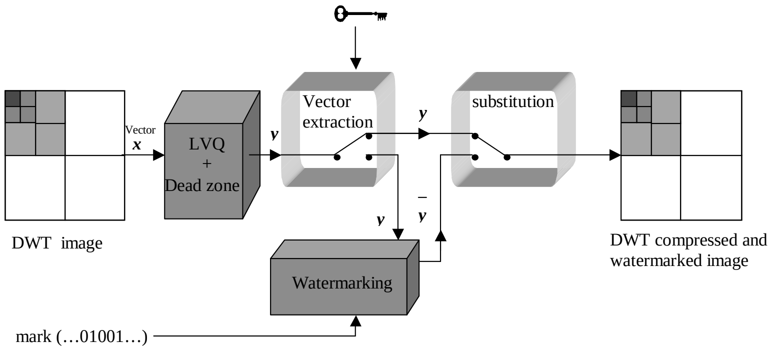

Figure 1: combined LVQ with dead zone + Watermarking scheme

$I(.)$ the indexing operation, and $Q(.)$ the quantization itself. We have $Q(\frac{x}{\gamma}) = \mathbf{y}$.The binary words associated to a lattice vector $\mathbf{y}$ are constructed here using an efficient prefix code which is based on the fact that $\mathbf{y}$ can be uniquely represented by a pair (e, p), where e is the index of the norm $\|\mathbf{y}\| = R$, and p the index which stands for the position of $\mathbf{y}$ on the shell of radius R.

Note that all the lattice vectors which belong to the shell of constant radius R have the same codeword length. Furthermore for efficiency in terms of bit rate, e is encoded using an entropy coding while p is only translated in binary [9].

2.2 LVQ with dead zone design

In this section, we introduce the concept of lattice vector quantizer with dead zone represented in figure 2. Let C be a pyramidal codebook in the lattice Z^n, with truncation radius R_t. The dead zone is dimension n is an hyper-pyramid of radius R_{DZ} $(0 < R_{DZ} \leq R_t)$ defined by:

$$DZ = \left\{ \mathbf{x} \in R^n | \, \|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i| \leq R_{DZ} \right\} \qquad (1)$$

The rules of both quantization and indexing are the same as those described in the previous section, except for source vectors belonging to DZ which are all quantized by the null vector. Because of its shape, the dead zone enables to decrease the coding bit rate while keeping the distortion constant (or equivalently to decrease the distortion for a fixed rate). Indeed, it better distributes the allocated bits on vectors with high energy by thresholding the low energy vectors, and permits to improve the visual quality on the decompressed image.

In this paper, the dead zone is designed for watermarking robustness improvement, as both selection, insertion and extraction steps are based on discrimination of null vectors.

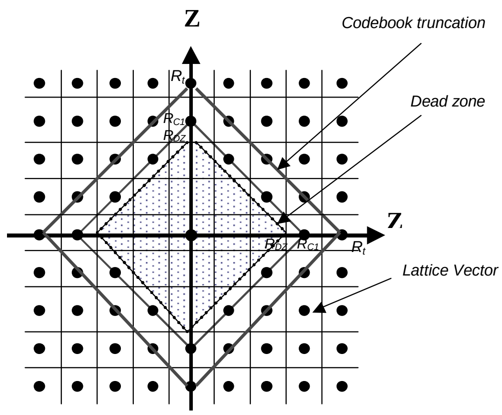

Figure 2: pyramidal DZLVQ

3 Watermarking algorithm

The main advantage of combining compression and watermarking stands in the fact it is possible to take into account the effects of compression when embedding the mark. Indeed, using a good bit allocation procedure (in both rate distortion sense and HVS characteristics)[10][3], DZLVQ tends to keep the most significant blocks of coefficients which represent robust candidates to carry a mark. Among all the available quantized DWT subbands, a good trade-off consists in working on the average resolutions. Indeed, on one hand, high resolutions components can be still attacked by postprocessing or pyracy actions, and on the other hand, modification on low resolution data can involve important degradation on the final visual quality. Typically, in a classical 5 level decomposition, we work on level 3 (and sometimes also level 2). In our scheme, the secret key takes also into account the position of vectors in these subbands and retains those only which are **non null**.

3.1 Insertion

Let M be a transform which embeds one bit per selected vector, and C^*_{DZ} the lattice code book with dead zone (without the null vector):

$$M : C^*_{DZ} \times \{0,1\} \rightarrow C^*_{DZ}(0) \cup C^*_{DZ}(1) \\ M(\mathbf{y}, i) \mapsto \overline{\mathbf{y}} \in C^*_{DZ}(i) \quad (2)$$

where i is the bit to embed, $\mathbf{y}$ the selected code book vector, $\overline{\mathbf{y}}$ the corresponding marked vector, and $C^*_{DZ}(i)$ the class which contains code book vectors carrying the bit i and defined by:

- vectors belonging to $C^*_{DZ}(0)$ are such that their two closest coordinates y_j and y_k $(j < k)$ verify $y_j < y_k$,
- vectors belonging to $C^*_{DZ}(1)$ are such that their two closest coordinates y_j and y_k $(j < k)$ verify $y_j > y_k$.

For example, $\mathbf{y} = (1, 8, 3, 5)$ belongs to $C^*_{DZ}(0)$ because $d(1, 3) = |1 - 3| = 2$ is minimum and $1 < 3$.

Furthermore, in order to minimize the visual impact of watermarking, we impose that M keeps the norm (or energy) of vectors constant:$\|\overline{\mathbf{y}}\| = \|\mathbf{y}\|$.

The proposed algorithm of insertion is based on the permutation of the two closest coordinates of the selected vector according to the bit i to embed. It runs as follows:

Algorithm 1

0- $j = 1$

1- $\overline{\mathbf{y}} = \mathbf{y}$

2- find k $(j < k)$ *such that* $d(y_j, y_k) = |y_j - y_k|$ *is minimum and non null*

3- permute y_j *and* y_k *in* $\overline{\mathbf{y}}$ *in order to have:*

$y_j < y_k$ *if i=0*

$y_j > y_k$ *if i=1*

4- read the embedded bit i' from $\overline{\mathbf{y}}$:

find the class $C^*_{DZ}(i')$ *to which* $\overline{\mathbf{y}}$ *belongs*

5- if $i' \neq i$*:* $j = j + 1$*: go to step 1*

else stop.

3.2 Extraction and robustness

The extraction scheme is based on two stages:

- *Detection*: selecting the vectors supposed to be marked according to the secret key (in our case, these vectors are sorted in the appropriate subbands of the DWT of the watermarked image),
- *Reading*: reading of the mark embedded in the selected vectors (step 4 of the insertion algorithm).

If the watermarked image has been corrupted by some post-processing or other attack, problems can occur in the two previous stages. Indeed for example, some vectors initially belonging to the dead zone can become non null and thus can be misincluded in the extraction process.

Suppose now the marked image has been attacked by a gaussian white noise with zero mean and standard deviation σ. We are going to calculate the maximum value of σ for good detection (with an error tolerance ε).

According to subsection 2.1, let $\overline{\mathbf{x}} = (k_1\gamma, ..., k_n\gamma)$ with $k_i \in Z$ be a decompressed marked vector in a subband J and $\tilde{\overline{\mathbf{x}}}$ the corresponding noisy vector. Figure 3 represents a zoom on the null vector discrimination area. We can easily see on this figure, that vectors $\tilde{\overline{\mathbf{x}}}$ which can be detected as marked vectors verify $\left\|Q(\tilde{\overline{\mathbf{x}}}))\right\|_1 \geq R_{C_1}$. On the contrary, vectors $\tilde{\overline{\mathbf{x}}}$ such that $\left\|Q(\tilde{\overline{\mathbf{x}}}))\right\|_1 \leq R_{C_{-1}}$ belong to the dead zone and must be removed of the extraction process.

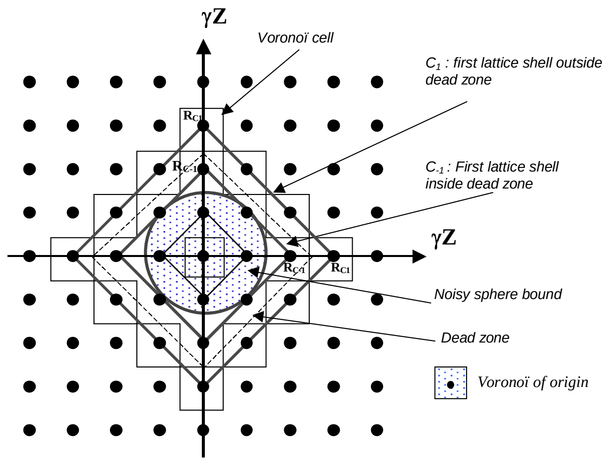

Figure 3: null vector discrimination area

Now let us examine the acceptable limit of the noise for correct detection, as represented in figure 3.We call detection failure the fact that at least one vector initially quantized by null vector is selected as a marked vector. Let $\mathbf{b}$ be a gaussian noise vector with n i.i.d. coordinates (zero mean and standard deviation σ). Gibson [7] gives the corresponding law for the radius of such vectors: $f_{\mathbf{r}}(r) = \frac{1}{2^{\frac{n-2}{2}}\Gamma\left(\frac{n}{2}\right)\sigma^n} r^{n-1} e^{-\frac{r}{2\sigma^2}}$, where $\mathbf{r} = \|\mathbf{b}\|_2$. Let z be the radius of the noisy sphere bound defined in figure 3: $z = \frac{R_{C_{-1}}\gamma}{\sqrt{2}}$ with $R_{C_{-1}} = \left\lfloor \frac{R_{DZ}}{\gamma} \right\rfloor$. Setting $A = \frac{z}{\sqrt{2}\sigma}$, we can write: $P\left(\|\mathbf{b}\|_2 < z\right) = \left(\frac{1}{2^{\frac{n-2}{2}}\Gamma\left(\frac{n}{2}\right)\sigma^n}\right)\left(\sqrt{2}\sigma\right)^n \frac{1}{2}\int_0^{\sqrt{A}} r^{\frac{n}{2}-1} e^{-r} dr$.

By successive integrations by parts, and setting $k = \frac{n}{2}, k \in N$,we finally obtain:

$$P\left(\|\mathbf{b}\|_2 < z\right) = 1 - e^{-\sqrt{A}} \sum_{i=0}^{k-1} \frac{1}{\Gamma(i+1)} \left(\sqrt{A}\right)^i \quad (3)$$

From formula 3, it is easy to calculate σ, since values of $P\left(\|\mathbf{b}\|_2 < z\right)$ can be pre-computed for a gaussian noise. Let us take for example: $R_{DZ} = 3, n = 8, \gamma = 6$ (which are typical values).The maximal noise standard deviation for a good detection with a risk $\varepsilon = 5\%$ is then $\sigma_0 \simeq 4.8$. For comparison, using the same parameters in the corresponding scheme without dead zone we obtain $\sigma_0 = 1.1$.Equivalently, to reach $\sigma_0 \simeq 4.8$ we need a large scaling factor $\gamma \simeq 25.9$. This is explained geometrically in figure 3. Indeed, for a scheme based on classical LVQ (without dead zone) the acceptable domain of noise for correct detection is reduced to the Voronoï of origin. It is thus obvious that dead zone improves the detection step and prevents extraction from desynchronization errors.

4 Simulation results and conclusion

Tests have been performed on lena image with a compression ratio of 1:64 (i.e. 0.125 bits per pixel). 5 level dyadic WT (with 9-7 filter) has been used and the mark has been embedded in the third resolution subbands.

Figure 4: DWT+DZLVQ compressed imge (compression ratio 1:64)

Figure 5: 60 bit message embedded (PSNR=30.87 dB)

Figures 4 and 5 show good performance in terms of invisibility. Furthermore, three kinds of tests have been performed using Matlab, and Stirmark attacks [8] to evaluate the robustness (for a 60 bit embedded message in 1:64 compressed image lena). Additionnal JPEG compression does not corrupt the mark until JPEG 25 quality (for JPEG 20 we have 2% of error). The mark is recovered without error for 3x3 median filtering, as well as gaussian white noise with zero mean and standard deviation of 7, in the spatial domain, which seems to be a limit in terms of visual quality.

Null vector discrimination is the crucial part of the detection step of our algorithm. In that sense, the dead zone offers more fiability to this step, according to the energy required to corrupt a null quantized vector. Furthermore, the combined approach allows to take into account the effects of compression when embedding the mark and seems to be of interest according to the role of both compression and watermarking in digital communications.

References

[1] M. Barlaud, M. Antonini, P. Mathieu, T. Gaidon: Wavelet and Image Communication, Editor Elsevier Jan Biemond series, 1994.

[2] J.H. Conway and N.J.A. Sloane: Sphere packing, lattices and group, Springer Verlag, 1988.

[3] J.F Delaigle, C. De Vleeschouwer, B. Macq: Watermarking algorithm based on a humain visual model, Signal Processing 66,1998.

[4] T.R. Fischer and J. Pan: Enumeration encoding and decoding algorithms for pyramid cubic lattice and trellis codes, IEEE Transactions on Information Theory, vol. 41, pp. 2056-2061, november 1995.

[5] Gerhard C. Langelaar, Iwan Setyawan, Reginald L. Lagendijk: Watermarking Digital Image and Video Data, A State-Of-The-Art Overview, IEEE Signal Processing Magazine, 2000.

[6] A. Gersho, R.M. Gray, "Vector Quantization and Signal Compression". Kluwer Academic Publishers, 1992.

[7] D.G. Jeong and J.D. Gibson: Uniform and piecewise uniform lattice vector quantization for memoryless gaussian and laplacian sources, IEEE Transactions on Information Theory, vol. 39, 3, pp. 786-804, may 1993.

[8] M. Kutter, F.A.P. Petitcolas: Fair evaluation methods for image watermarking systems, Journal of Electronic Imaging, 9(4), 2000.

[9] J.M. Moureaux, P. Loyer and M. Antonini: Low-Complexity Indexing Method for Zn and Dn lattice quantizers, IEEE Transactions on Communications, vol. 46, 12, pp. 1602-1609, december 1998.

[10] P. Raffy, M. Antonini, and M. Barlaud: A New Optimal Subband Bit Allocation Procedure for Very Low Bit Rate Image Coding, IEE Electronics Letters, vol.34, issue 7, pp. 647, April 1998.