ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2011, Vol. 40, No. 2

TWO-DIMENSIONAL VECTOR QUANTIZER WITH VARIABLE LENGTH LOSSLESS CODE FOR LAPLACIAN SOURCE

Milan R. Dincic, Zoran H. Peric

Faculty of Electronic Engineering, University of Nis
Aleksandra Medvedeva 14, 18000 Nis, Serbia
e-mail: mdincha@hotmail.com, zoran.peric@elfak.ni.ac.rs

crossref http://dx.doi.org/10.5755/j01.itc.40.2.428

Abstract. The main aim of this paper is to apply variable length lossless code on output points of the two-dimensional vector quantizer for Laplacian source. This problem is not yet solved in the literature. In this paper two-dimensional quantizer is designed using Helmert transform and its optimization is done. New lossless code is introduced. It is very simple but very close to the ideal code since it gives the bit-rate very close to the entropy. It is applied on the two-dimensional quantizer. It is shown that our two-dimensional quantizer can achieve the same performances as scalar uniform and nonuniform quantizers, but using much smaller number of points per dimension. Therefore, two-dimensional quantizer has smaller execution complexity.

Keywords: vector quantizer, lossless code, Golomb-Rice code, execution complexity.

1. Introduction

Quantization is very important step in the process of analog to digital conversion. Quantization is used in almost all modern telecommunication systems. Good choice of quantizer can increase signal quality and decrease bit-rate.

Quantizers can be scalar and vector. Design of some types of scalar quantizers were analyzed in [1, 2, 3]. Vector quantizers give better performances (higher signal-to-quantization noise ratio (*SQNR*) for the same bit-rate) compared to scalar quantizers, since they have a higher degree of freedom for choosing the reconstruction values and the decision regions [4]. But, vector quantizers are, in general case, more complex than scalar quantizers, and their complexity increases with increase of quantizer dimension [4]. Two-dimensional quantizers are the simplest, and therefore the most often used vector quantizers.

Lossless compression codes can be used for coding of symbols of discrete sources, decreasing bit-rate. Representation points of quantizer can be considered as symbols of discrete source, and therefore they can be coded with lossless codes. Huffman code is the most popular lossless code [5, 6, 7, 8, 9]. But, the complexity of Huffman code drastically increases when the number of symbols increases, since it is needed to form code tree in coder, as well as in decoder. Also, probability for each symbol has to be calculated. Because of that, Huffman code is inapplicable for sources with large number of symbols. Huffman code can be used for coding of levels of scalar quantizer, if the number of levels is not too large (for example, few tens). But, the number of points of two-dimensional quantizers can be very large. Because of that, Huffman code is practically inapplicable for scalar quantizers with large number of levels and for two-dimensional quantizers.

A simple lossless code was proposed in [10] and it was applied on scalar, but not on vector quantizers. Another important lossless code is Golomb-Rice code. It is much simpler for realization than Huffman code, since this code is computed with a few logical operations and there is no need to form code tree [11, 12, 13]. This feature is especially important for the decoder, which has to be simple and fast. It was proved that Golomb-Rice code is especially suitable for Laplacian source. Due to all these facts, Golomb-Rice code is included in many modern compression standards, such as JPEG-LS [14], MPEG-4 ALS (Audio Lossless Coding) [15], and CCSDS (Consultative Committee for Space Data Systems) recommendation for lossless data compression [16]. Also, Golomb-Rice code can be used for compression of test data [17]. Golomb-Rice code was applied on scalar quantizer in [18, 19]. But, application of Golomb-Rice code on vector quantizers has not been analized by now, as we know. So, we can conclude that application of lossless code on vector quantizer is an open problem, not yet solved in the literature. In this paper we propose a solution for this problem.

The main aim of this paper is application of lossless code for coding representation points of two-dimensional quantizer. Firstly, design of two-dimensional quantizer is done for memoryless Laplacian source, using Helmert transform [20]. The quantizer is designed in the way that one coordinate (denoted with *r*) is quantized using hybrid quantizer (combination of uniform and nonuniform quantizers) and the other coordinate (denoted with *u*) is quantized using uniform quantizer. It is proved in this paper that hybrid quantizer is the best solution for *r* coordinate, better than uniform and nonuniform quantizers. The complexity of the hybrid quantizer is between complexities of the uniform and the nonuniform quantizers. Quantizer's cells are placed in concentric rectangular rings. Optimization of the number of cells in each ring is done. After that, a new lossless code is defined. This lossless code is a modification of Golomb-Rice code, but it is more flexible than Golomb-Rice code. Our code gives the bit-rate which is very close to the entropy of the source. The main advantage of our lossless code is its simplicity. It is drastically simpler than Huffman code since there is no need to form a code tree and also there is no need to calculate probabilities of representation points. Therefore, the lossless code can be very easily applied on two-dimensional quantizers with large number of points. It is shown in the paper that two-dimensional quantizer with variable length codewords can achieve much better performances, compared to scalar uniform and nonuniform quantizers, using the same number of points per dimension (or, in other words, our model can achieve the same performances using much smaller number of points per dimension). For example, our model can satisfy the G.712 standard using in average 92 points per dimension, which is much smaller compared to 128 points needed for scalar nonuniform quantizer or 256 points needed for scalar uniform quantizer. Therefore, our model has smaller execution complexity, and it is faster.

This paper is organized as follows. In Section 2, design of two-dimensional quantizer is described using Helmert transform, expressions for distortion are derived, optimal numbers of cells in rings are calculated and the lossless code is described. Section 3 gives some numerical results. In Section 4, some additional explanations are given and execution complexity of our model is discussed. Section 5 concludes the paper.

2. Design of the two-dimensional quantizer with the lossless code

2.1. Helmert transform

The 2-D (two-dimensional) probability density function for independent identically distributed Laplace random variables (source) with zero mean and the unity variance is given as

$$f(\mathbf{x}) = \frac{1}{2} e^{-\sqrt{2}\left(|x_1|+|x_2|\right)}, \qquad (1)$$

where $\mathbf{x}$ is the source vector with elements x_1 and x_2. To simplify the vector quantizer, the Helmert transform [20] is applied on the source vector. This transform is defined as mapping $(x_1, x_2, S) \to (r, u, A)$, so that

$$r = \frac{1}{\sqrt{2}}\left(|x_1| + |x_2|\right);\, u = \frac{1}{\sqrt{2}}\left(|x_1| - |x_2|\right);\, A = S\,, \qquad (2)$$

where S represents the quadrant where the point (x_1, x_2) is placed. S can take values from the set $\{1, 2, 3, 4\}$. These quadrants are presented in Figure 1(left). A represents the area where the point (r, u) is placed, and A also can take values from the set $\{1, 2, 3, 4\}$. These areas are presented in Figure 1(right). Point (x_1, x_2) from the quadrant i is mapped into point (r, u) in the area i, $i = 1, 2, 3, 4$. These additional information S and A about quadrants in (x_1, x_2) domain and about areas in (r, u) domain are needed to provide one-to-one mapping, i.e. to provide uniquely defined transform. Based on the expression (2) and Figure 1b, we can conclude that $0 \le r < +\infty$ and $-r \le u \le r, \forall r$.

Effect of Helmert transform is rotation of the coordinate system for 45 degrees clockwise, i.e. (r, u) coordinate system is obtained by rotation of (x_1, x_2) coordinate system for 45 degrees clockwise. This is shown in Figure 2.

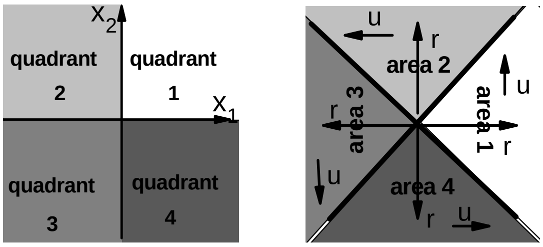

Figure 1. Quadrants in the (x_1, x_2) domain (left) and areas in the (r, u) domain (right)

Inverse mapping $(r, u, A) \to (x_1, x_2, S)$ is defined as:

$$|x_1| = \frac{1}{\sqrt{2}}(r + u);\, |x_2| = \frac{1}{\sqrt{2}}(r - u);\, S = A\,. \qquad (3)$$

Based on S and using (3), x_1 and x_2 are calculated in the following way: $x_1 = |x_1|$ if $S \in \{1, 4\}$; $x_1 = -|x_1|$ if $S \in \{2, 3\}$; $x_2 = |x_2|$ if $S \in \{1, 2\}$; $x_2 = -|x_2|$ if $S \in \{3, 4\}$.

The probability density function defined with (1), in (r,u) domain becomes

$$f(r,u)=\frac{1}{2}e^{-2r}. \qquad (4)$$

Probability density function for r coordinate is

$$f(r)=4\int_{u=-r}^{r}f(r,u)du=4\int_{u=-r}^{r}\frac{1}{2}e^{-2r}du=4re^{-2r}. \quad (5)$$

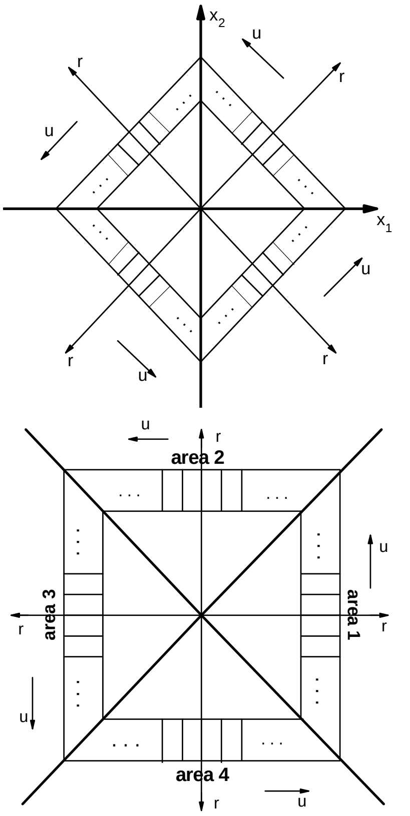

Figure 2. Due to Helmert transform, (r,u) coordinate system is obtained by rotation of (x_1,x_2) coordinate system for 45 degrees clockwise

2.2. Two-dimensional (2D) quantizer

Now, we will define two-dimensional quantizer in (r,u) domain. The total number of cells of that quantizer is *N*. For r coordinate, hybrid quantizer will be applied. For u coordinate, uniform quantizer will be applied.

Quantization of r **coordinate**. As it was said, hybrid quantizer is used for quantization of r coordinate. This hybrid quantizer is combination of a uniform and a nonuniform companding quantizers. r^* denotes the value of r coordinate which is the border between uniform and nonuniform quantizers. Uniform quantizer consists of L_1 levels for r coordinate. Thresholds for this uniform quantizer are denoted by $r_i, i=0,...,L_1$ and they can be calculated as $r_i=ir^*/L_1$. It is valid that $r_0=0$ and $r_{L_1}=r^*$. Representation levels for uniform quantizers are $m_i=(i-1/2)r^*/L_1, i=1,...,L_1$. Nonuniform quantizer for r coordinate is companding quantizer with optimal compression function $g(r):(r^*,\infty)\to(0,1)$, which is defined as [1]:

$$g(r)=\frac{\int_{r^*}^{r}\sqrt[3]{f(r)}dr}{\int_{r^*}^{\infty}\sqrt[3]{f(r)}dr}. \qquad (6)$$

For $f(r)$ defined with (5), compression function $g(r)$ becomes

$$g(r)=\frac{1}{2r^*\,\mathrm{E}\left(-\frac{1}{3},\frac{2r^*}{3}\right)}\left[3e^{-2r^*/3}+\mathrm{E}\left(\frac{2}{3},\frac{2r^*}{3}\right)\right.$$
$$\left.-\left(\frac{r}{r^*}\right)^{1/3}\left(3e^{-2r/3}+\mathrm{E}\left(\frac{2}{3},\frac{2r}{3}\right)\right)\right], \qquad (7)$$

where $\mathrm{E}(a,z)=\int_{1}^{+\infty}e^{-zt}t^{-n}dt$ is exponential integral function, which is very suitable for numerical computation. Nonuniform companding quantizer has L_2 levels. Thresholds for this nonuniform quantizer are denoted by $r^*=r_{L_1}<r_{L_1+1}<...<r_{L_1+L_2-1}<r_{L_1+L_2}=+\infty$. Representation levels are denoted by $m_{L_1+1}<m_{L_1+2}<...<m_{L_1+L_2}$. Thresholds and representation levels for nonuniform quantizer can be found by numerical solving of the following equations:

$$g(r_i)=\frac{i-L_1}{L_2}, i=L_1+1,...,L_1+L_2-1, \qquad (8)$$

$$g(m_i)=\frac{1}{L_2}\left(i-L_1-\frac{1}{2}\right),\ i=L_1+1,...,L_1+L_2\ .(9)$$

There are totally L_1+L_2 levels for r coordinate. Each level for r coordinate makes one ring in two-dimensional plane. Therefore, we have totally L_1+L_2 concentric rings. The area which consists of the first L_1 rings (where r coordinate is uniformly quantized) is called uniform part of 2D quantizer, and the area which consists of L_2 rings where r coordinate is nonuniformly quantized is called nonuniform part of 2D quantizer.

Quantization of u **coordinate**. Inside each ring, uniform quantization of u coordinate is done. On the *i*-th ring we have M_i levels for u coordinate, $M_i/4$ in each of four areas. Thresholds for u coordinate on the i-th ring ($i = 1,...,L_1 + L_2$) are denoted by $u_{i,j}$, $j = 0,..., M_i$ and representation levels are denoted by $\hat{u}_{i,j}$, $j = 1,...,M_i$. For rings $i = 1,...,L_1 + L_2 - 1$, quantization cells $S_{i,j}\left([r_{i-1}, r_i);[u_{i,j-1}, u_{i,j})\right)$ are rectangular, which is shown in Figure 3 (up). The quantization stepsize for u coordinate on the i-th ring, $i = 1,..., L_1 + L_2 - 1$, is

$$\Delta u_i = \frac{r_{i-1} + r_i}{M_i / 4} = \frac{4(r_{i-1} + r_i)}{M_i}. \qquad (10)$$

On the ($L_1 + L_2$)-th ring, quantization cells $S_{L_1+L_2,j}\left([r_{L_1+L_2-1}, \infty);[u_{L_1+L_2,j-1}, u_{L_1+L_2,j})\right)$ are also rectangular, but unbounded on the one side. This is shown in Figure 3 (down). For the ($L_1 + L_2$)-th ring, the quantization stepsize for u coordinate is

$$\Delta u_{L_1+L_2} = \frac{2r_{L_1+L_2-1}}{M_{L_1+L_2} / 4} = \frac{8r_{L_1+L_2-1}}{M_{L_1+L_2}}. \qquad (11)$$

Within each cell $S_{i,j}$ there is one representation point $(m_i, \hat{u}_{i,j})$. There are four overload cells which are presented in Figure 3 (down). Within each of these four overload cells there is one representation point, which will be defined later. The total number of quantization cells of two-dimensional quantizer is denoted with $N = N_1 + N_2$. $N_1 = \sum_{i=1}^{L_1} M_i$ is the total number of cells in the uniform part, and $N_2 = \sum_{i=L_1+1}^{L_1+L_2} M_i + 4$ is the total number of cells in the nonuniform part. Additional four cells in the expression for N_2 are overload cells.

2.3. Distortion

The total distortion D consists of three parts, i.e., $D = D_1 + D_2 + D_3$. D_1 is distortion in the first $L_1 + L_2 - 1$ rings where cells are bounded; D_2 is distortion in the ($L_1 + L_2$)-th ring, where cells are unbounded; D_3 is distortion in four overload cells. This is shown in Figure 4. All these distortions (D, D_1, D_2 and D_3) are distortions per dimension. Usually, for vector quantization, distortion is given per dimension, to simplify comparison of performances of vector quantizers with different dimensions.

Distortion D_1 is given with the following expression:

$$D_1 = \frac{1}{2}\sum_{i=1}^{L_1+L_2-1}\sum_{j=1}^{M_i}\int_{r_{i-1}}^{r_i}\int_{u_{i,j-1}}^{u_{i,j}}\left[(r - m_i)^2 + (u - \hat{u}_{i,j})^2\right]\frac{1}{2}e^{-2r}\,du\,dr. \qquad (12)$$

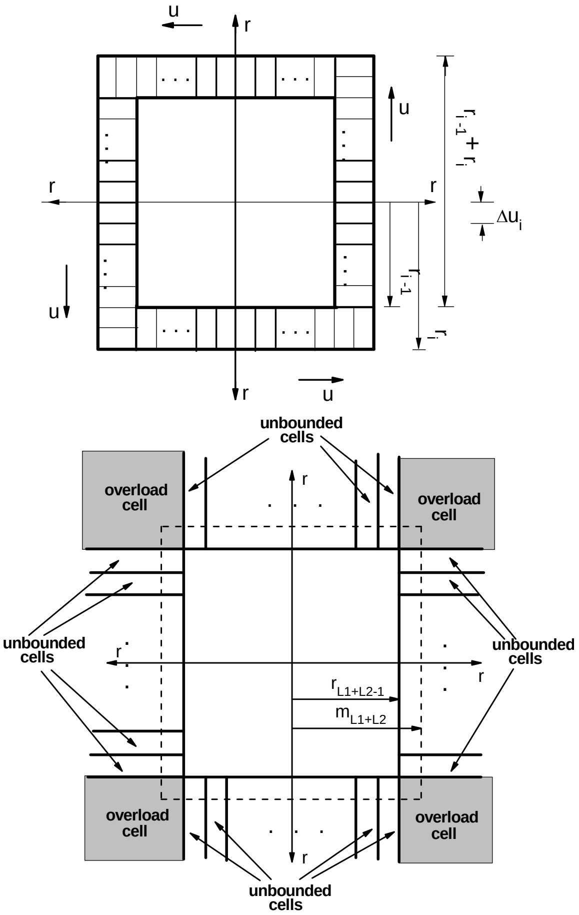

Figure 3. Rectangular cells on the i-th ring, $i = 1,..., L_1 + L_2 - 1$ (up); unbounded rectangular cells and four overload cells on the ($L_1 + L_2$)-th ring (down)

The factor 1/2 at the beginning of the expression (12) means that D_1 is distortion per dimension. After integration by u it is obtained:

$$D_1 = \frac{1}{4}\sum_{i=1}^{L_1+L_2-1}\sum_{j=1}^{M_i}\int_{r_{i-1}}^{r_i}\left[(r - m_i)^2\Delta u_i + \frac{\Delta u_i^3}{12}\right]e^{-2r}dr$$

$$= \frac{1}{4}\sum_{i=1}^{L_1+L_2-1} M_i \int_{r_{i-1}}^{r_i}\left[(r - m_i)^2\Delta u_i + \frac{\Delta u_i^3}{12}\right]e^{-2r}dr. \qquad (13)$$

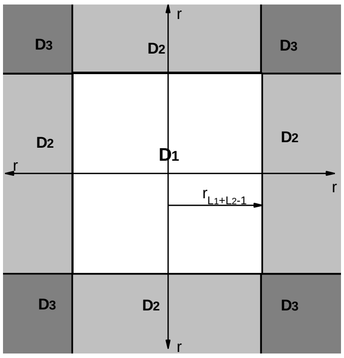

Figure 4. Regions where distortions D_1, D_2 and D_3 are defined

For Δu_i defined with (10), we obtain:

$$D_1 = \sum_{i=1}^{L_1+L_2-1} (r_{i-1}+r_i) \int_{r_{i-1}}^{r_i} \left[(r-m_i)^2 + \frac{4(r_{i-1}+r_i)^2}{3M_i^2} \right] e^{-2r} dr \, . \tag{14}$$

Finally, after basic integration, the following expression is obtained:

$$D_1 = \sum_{i=1}^{L_1+L_2-1} \left((r_{i-1}+r_i) \left[e^{-2r_{i-1}} \left(\frac{2}{3} \frac{(r_{i-1}+r_i)^2}{M_i^2} + \frac{1}{4}\left(1+2(r_{i-1}-m_i)(1+r_{i-1}-m_i)\right) \right) - e^{-2r_i} \left(\frac{2}{3} \frac{(r_{i-1}+r_i)^2}{M_i^2} + \frac{1}{4}\left(1+2(r_i-m_i)(1+r_i-m_i)\right) \right) \right] \right) . \tag{15}$$

Previous expression (15) can be written as $D_1 = \sum_{i=1}^{L_1+L_2-1}(*) = \sum_{i=1}^{L_1}(*) + \sum_{i=L_1+1}^{L_1+L_2-1}(*) = D_{11} + D_{12}$, where (*) denotes the expression under summation in (15). D_{11} denotes distortion in the uniform part (i.e. in the first L_1 rings) and using expressions for r_i and m_i which were defined in Subsection 2. 2., it is obtained that

$$D_{11} = \sum_{i=1}^{L_1} \left((2i-1)\frac{r^*}{L_1} \left[e^{-2(i-1)\frac{r^*}{L_1}} \left(\frac{1}{4}\left(1 - \frac{r^*}{L_1} + \frac{1}{2}\left(\frac{r^*}{L_1}\right)^2 \right) + \frac{2}{3} \frac{((2i-1)r^*/L_1)^2}{M_i^2} \right) - e^{-2i\frac{r^*}{L_1}} \left(\frac{1}{4}\left(1 + \frac{r^*}{L_1} + \frac{1}{2}\left(\frac{r^*}{L_1}\right)^2 \right) + \frac{2}{3} \frac{((2i-1)r^*/L_1)^2}{M_i^2} \right) \right] \right) . \tag{16}$$

D_{12} represents distortion in rings $L_1+1,\ldots,$ L_1+L_2-1; r_i and m_i defined with (8) and (9) should be used in calculation of D_{12} .

Distortion in the (L_1+L_2)-th ring where cells are unbounded, denoted as D_2 , is defined as:

$$D_2 = \frac{1}{2} \sum_{j=1}^{M_{L_1+L_2}} \int_{r_{L_1+L_2-1}}^{\infty} \int_{u_{L_1+L_2,\,j-1}}^{u_{L_1+L_2,\,j}} \left[\left(r - m_{L_1+L_2}\right)^2 + \left(u - \hat{u}_{L_1+L_2,\,j}\right)^2 \right] \frac{1}{2} e^{-2r} du dr = \frac{1}{4} \sum_{j=1}^{M_{L_1+L_2}} \int_{r_{L_1+L_2-1}}^{\infty} \left[\frac{\Delta u_{L_1+L_2}^3}{12} + \left(r - m_{L_1+L_2}\right)^2 \Delta u_{L_1+L_2} \right] e^{-2r} dr \, . \tag{17}$$

After integration and summation, and using (11), it is obtained:

$$D_2 = r_{L_1+L_2-1} e^{-2r_{L_1+L_2-1}} \left[\frac{16 r_{L_1+L_2-1}^2}{3M_{L_1+L_2}^2} + \frac{1}{2}(1+2(r_{L_1+L_2-1} - m_{L_1+L_2})(1+r_{L_1+L_2-1} - m_{L_1+L_2})) \right] . \tag{18}$$

The overload distortion D_3 represents distortion in four overload cells. In each overload cell there is one representation point. To minimize distortion, this representation point should be placed on the bisector of the overload cell. Therefore, values of *r* and *u* coordinates of the representation point are equal and denoted with *c*. The overload distortion D_3 can be calculated as $D_3 = 8 \cdot D_3^{\text{half}}$, where $D_3^{\text{half}} = \frac{1}{2} \int_{r_{L_1+L_2-1}}^{\infty} \int_{u=r_{L_1+L_2-1}}^{r} \left[(r-c)^2 + (u-c)^2 \right] \frac{1}{2} e^{-2r} du dr$ represents distortion in the half of the one overload cell. This is shown in Figure 5.

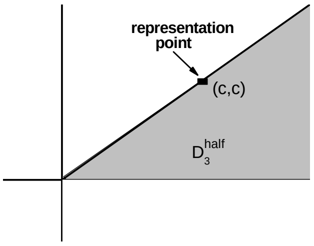

Figure 5. One overload cell with the representation point placed on the bisector of the overload cell

After integration, it is obtained that

$$D_3 = e^{-2r_{L_1+L_2-1}} \left[\frac{7}{4} + r_{L_1+L_2-1}^2 + 2r_{L_1+L_2-1} + c^2 - 2c - 2cr_{L_1+L_2-1} \right] . \tag{19}$$

Minimizing D_3 , $\frac{\partial D_3}{\partial c} = 2c - 2 - 2r_{L_1+L_2-1} = 0$, we obtain the optimal value for parameter *c*

$$c = 1 + r_{L_1+L_2-1} \, . \tag{20}$$

Changing this into (20), we obtain the following expression for the overload distortion

$$D_3 = \frac{3}{4} e^{-2r_{L_1+L_2-1}} \, . \tag{21}$$

The total distortion *D* is equal to the sum of expressions (15), (18) and (21). Signal-to-quantization noise ratio (*SQNR*), for unit variance, is defined as:

$$SQNR \text{ [dB]} = 10 \log_{10} \frac{1}{D} \, . \tag{22}$$

2.4. Optimal number of cells on rings

In this subsection, optimal values of the number of cells on each ring, denoted with $M_i, i = 1,\dots, L_1 + L_2$, will be found, in the way to minimize distortion *D*. Two condition must be fulfilled: $\sum_{i=1}^{L_1} M_i = N_1$ and $\sum_{i=L_1+1}^{L_1+L_2} M_i = N_2 - 4$. So, minimization of *D* with these constraints should be done, and this will be done using the technique of Lagrange multipliers. Firstly, we define function *J* in the following way:

$$J = D + \lambda_1 \left(\sum_{i=1}^{L_1} M_i - N_1 \right) + \lambda_2 \left(\sum_{i=L_1+1}^{L_1+L_2} M_i - (N_2 - 4) \right), \qquad (23)$$

where λ_1 and λ_2 are Lagrange multipliers. Now, minimization of *J* will be done.

For $M_i, i = 1,\dots, L_1$ it is valid that

$$\frac{\partial J}{\partial M_i} = \frac{\partial D}{\partial M_i} + \lambda_1 = \frac{\partial D_{11}}{\partial M_i} + \lambda_1 = \frac{-4}{3M_i^3}\left((2i-1)\frac{r^*}{L_1} \right)^3 \left(e^{-2(i-1)\frac{r^*}{L_1}} - e^{-2i\frac{r^*}{L_1}} \right) + \lambda_1 = 0. \qquad (24)$$

From (24) it follows that

$$M_i = \frac{1}{\sqrt[3]{\lambda_1}} (2i-1) \frac{r^*}{L_1} e^{-2i\frac{r^*}{3L_1}} \sqrt[3]{\frac{4}{3}\left(e^{2\frac{r^*}{L_1}} - 1 \right)},$$
$$i = 1,\dots, L_1 .(25)$$

Changing these values in the condition $\sum_{j=1}^{L_1} M_j = N_1$, it is obtained that

$$\frac{1}{\sqrt[3]{\lambda_1}} = \frac{N_1}{\frac{r^*}{L_1} \sqrt[3]{\frac{4}{3}\left(e^{2\frac{r^*}{L_1}} - 1 \right)} \sum_{j=1}^{L_1} (2j-1) e^{-2j\frac{r^*}{3L_1}}}. \qquad (26)$$

Changing this in expression (25), we obtain expressions for the optimal values of the numbers of cells in rings in the uniform part:

$$M_i = N_1 \frac{(2i-1) e^{-2i\frac{r^*}{3L_1}}}{\sum_{j=1}^{L_1} (2j-1) e^{-2j\frac{r^*}{3L_1}}}, i = 1,\dots, L_1. \qquad (27)$$

For $M_i, i = L_1 + 1,\dots, L_1 + L_2 - 1$ it is valid that

$$\frac{\partial J}{\partial M_i} = \frac{\partial D}{\partial M_i} + \lambda_2 = \frac{\partial D_{12}}{\partial M_i} + \lambda_2 = -\frac{4}{3M_i^3} (r_{i-1} + r_i)^3 \left(e^{-2r_{i-1}} - e^{-2r_i} \right) + \lambda_2 = 0. \qquad (28)$$

From the previous expression it follows that

$$M_i = \frac{1}{\sqrt[3]{\lambda_2}} (r_{i-1} + r_i) \sqrt[3]{\frac{4}{3}\left(e^{-2r_{i-1}} - e^{-2r_i} \right)},$$
$$i = L_1 + 1,\dots, L_1 + L_2 - 1. \qquad (29)$$

For $M_{L_1+L_2}$ it is valid that

$$\frac{\partial J}{\partial M_{L_1+L_2}} = \frac{\partial D}{\partial M_{L_1+L_2}} + \lambda_2 = \frac{\partial D_2}{\partial M_{L_1+L_2}} + \lambda_2 = -\frac{32}{3M_{L_1+L_2}^3} r_{L_1+L_2-1}^3 e^{-2r_{L_1+L_2-1}} + \lambda_2 = 0. \qquad (30)$$

It follows that

$$M_{L_1+L_2} = \frac{2}{\sqrt[3]{\lambda_2}} r_{L_1+L_2-1} \sqrt[3]{\frac{4}{3} e^{-2r_{L_1+L_2-1}}}. \qquad (31)$$

Changing expressions (29) and (31) into condition $\sum_{j=L_1+1}^{L_1+L_2} M_j = N_2 - 4$, it is obtained that

$$\sqrt[3]{\lambda_2} = \frac{1}{N_2 - 4} \left[2r_{L_1+L_2-1} \sqrt[3]{\frac{4}{3} e^{-2r_{L_1+L_2-1}}} + \left(\sum_{j=L_1+1}^{L_1+L_2-1} (r_{j-1} + r_j) \sqrt[3]{\frac{4}{3}\left(e^{-2r_{j-1}} - e^{-2r_j} \right)} \right) \right]. \qquad (32)$$

Changing this into (29) and (31), we obtain optimal values of cells number for rings in the nonuniform part:

$$M_i = \frac{(N_2 - 4)(r_{i-1} + r_i)\sqrt[3]{e^{-2r_{i-1}} - e^{-2r_i}}}{\left(\sum_{j=L_1+1}^{L_1+L_2-1} (r_{j-1} + r_j)\sqrt[3]{e^{-2r_{j-1}} - e^{-2r_j}} \right) + 2r_{L_1+L_2-1}\sqrt[3]{e^{-2r_{L_1+L_2-1}}}},$$
$$i = L_1 + 1,\dots, L_1 + L_2 - 1, \qquad (33)$$

$$M_{L_1+L_2} = (N_2 - 4) 2r_{L_1+L_2-1} \sqrt[3]{e^{-2r_{L_1+L_2-1}}} \Big/ \Big[2r_{L_1+L_2-1} \cdot \sqrt[3]{e^{-2r_{L_1+L_2-1}}} + \left(\sum_{j=L_1+1}^{L_1+L_2-1} (r_{j-1} + r_j) \sqrt[3]{e^{-2r_{j-1}} - e^{-2r_j}} \right) \Big]. \qquad (34)$$

2.5. Regions making and modification of the number of cells

Lossless code will be applied for coding representation points of our two-dimensional quantizer. This lossless code will be described in Subsection 2.6. In this subsection some preparations for the application of the lossless code will be done. To apply the lossless code, we should make regions. Inside uniform part of the two-dimensional quantizer (the first L_1 rings), we make $S = L_1 / k$ regions, by joining k consecutive rings in one region. It is valid that the i-th region ($i = 1,...,S$) consists of rings $(i-1)k+1, (i-1)k+2,..., ik$. The whole nonuniform part (with L_2 rings and N_2 cells) is one, the ($S+1$)-th region. Four overload cells also belong to the ($S+1$)-th region. Therefore, we have totally $S+1$ regions. Q_i, $i = 1,..., S+1$ denotes the number of cells in the i-th region. It is valid that $Q_i = \sum_{j=(i-1)k+1}^{ik} M_j$ for $i = 1,..., S$, where M_j are given with (27). Also, it is valid that $Q_{S+1} = N_2$.

To apply the lossless code, one condition should be fulfilled: the number of cells within each region must be a power of two, i.e. $Q_i = 2^{\alpha_i}$, $i = 1,..., S+1$; $\alpha_i \in \mathrm{N}$. Firstly, we will consider regions in the uniform part. For M_j given with (27), this condition is not satisfied in general case. Therefore, we should modify the number of cells in regions in the uniform part. With $Q_i^* = 2^{[\log_2 Q_i]}$ is denoted the nearest number of Q_i which is the power of two. [*x*] is the nearest integer of the real number *x*. Now, instead of Q_i cells, each region in the uniform part has Q_i^* cells. Since the number of cells in regions is modified, the number of cells in rings also should be modified. The modified number of cells in the j-th ring, which is in the i-th region, is $M_j^* = M_j - (Q_i - Q_i^*)/k$, $i = 1,..., S$; $j = (i-1)k+1,..., ik$. So, inside the i-th region, the total number of cells is changed by $Q_i - Q_i^*$ cells, and this change is equally distributed on all k rings inside this region, i.e., the number of cells in each ring in this region is changed by $(Q_i - Q_i^*)/k$. It is valid that $Q_i^* = \sum_{j=(i-1)k+1}^{ik} M_j^*$. The modified number of cells within the uniform part is $N_1^* = \sum_{i=1}^{L_1} M_i^*$. So, on the beginning of the design process the cells number in the uniform part was set to the initial value N_1. This initial value N_1 is chosen to achieve required performances (for example, if the G.712 standard should be satisfied, *SNR* higher that 34 dB should be achieved). But during the design process this initial value is modified, and we obtain new number of cells in the uniform part, denoted with N_1^*. N_1^* is usually very close to the initial value N_1.

Since the whole nonuniform part is one region, the number of cells in the nonuniform part must be a power of two. If the initial number of cells N_2 is not a power of two, it should be changed with N_2^*, the nearest integer to N_2 which is a power of two. So, $Q_{S+1}^* = N_2^*$. Then, modified numbers of cells in rings in the nonuniform part are $M_j^* = M_j - (N_2 - N_2^*)/L_2$, $j = L_1+1,..., L_1+L_2$.

Now, modified values M_j^*, $j = 1,..., L_1 + L_2$ should be used in expressions (15) and (18), instead of initial values M_i.

2.6. Lossless code with the variable length codewords

In this section the lossless code will be described. As it was said, we have $S+1$ regions. Within the *i*-th region there are Q_i^* representation points (one point in each cell). One index ξ_i ($0 \le \xi_i \le Q_i^* - 1$) can be assigned to each point in the *i*-th region. Points are coded with codewords using the following coding rule:

Rule 1). Point from the *i*-th region, ($1 \le i \le S+1$) with index ξ_i is coded with codeword $\underbrace{1...1}_{i-1}0\ \underbrace{x...x}_{\log_2 Q_i^*}$, where $\underbrace{x...x}_{\log_2 Q_i^*}$ is the natural binary code of the index ξ_i. All points within one region are coded with codewords with the same lengths $l_i = i + \log_2 Q_i^*$.

This lossless code is a modification of Golomb-Rice code. In Golomb-Rice code [8] numbers of cells in each region should be the same, i.e. $Q_1^* = Q_2^* = ... = Q_{S+1}^*$. In our code, numbers of cells in different regions can be mutually different. Therefore, our code is more flexible than Golomb-Rice code and there are more possibilities to obtain required performances by adjusting the number of cells in regions.

Now, we will explain how to assign index ξ_i to points in some region. Let's consider the *i*-th ($1 \le i \le S$) region in the uniform part. In this region there are *k* rings: $(i-1)k+1,..., ik$. Index assignment procedure is defined with the following three rules:

Rule 2). Rings are coded in increasing order, i.e. ring $(i-1)k+1$ is coded firstly, after that ring $(i-1)k+2$,…, and finally ring ik. This means that the first $M_{(i-1)k+1}^*$ indexes ($0 \le \xi_i \le M_{(i-1)k+1}^* - 1$) are assigned to the points in the ring $(i-1)k+1$; the next $M_{(i-1)k+2}^*$ indexes ($M_{(i-1)k+1}^* \le \xi_i \le M_{(i-1)k+1}^* + M_{(i-1)k+2}^* - 1$) are assigned to the points in the ring $(i-1)k+2$;…; the last M_{ik}^* indexes ($\sum_{j=1}^{k-1} M_{(i-1)k+j}^* \le \xi_i \le Q_i^* - 1$) are assigned to the points in the ring ik.

Rule 3). Within each ring, points from Area 1 are coded firstly, after that points from Area 2, after that points from Area 3 and finally from Area 4. For example, in the ring $(i-1)k+1$, indexes $0 \le \xi_i \le M^*_{(i-1)k+1}/4-1$ are assigned to the points in Area 1, indexes $M^*_{(i-1)k+1}/4 \le \xi_i \le M^*_{(i-1)k+1}/2-1$ to the points in Area 2, indexes $M^*_{(i-1)k+1}/2 \le \xi_i \le 3M^*_{(i-1)k+1}/4-1$ to the points in Area 3 and indexes $3M^*_{(i-1)k+1}/4 \le \xi_i \le 3M^*_{(i-1)k+1}-1$ to the points in Area 4. Using this rule, information about the area where some point belongs is included in index (and therefore in the codeword) and there is no need to transmit additional information about the area. Based on the received codeword, receiver can determine to which area this point belongs. Recall that the receiver must know information about the area for calculation of the inverse Helmert transform (see Subsection 2.1.).

Rules 2) and 3) can be joined in the following rule:

Rule 2-3). Indexes which are assigned to the points in the *i*-th region, in the $(i-1)k+p$ ring, in the *m*-th area ($i=1,...,S$; $p=1,...,k$; $m=1,2,3,4$) should satisfy the following condition:

$$\begin{cases} (m-1)M^*_{(i-1)k+1}/4 \le \xi_i \le \\ mM^*_{(i-1)k+1}/4-1, \text{ for } p=1; \\ \sum_{j=1}^{p-1} M^*_{(i-1)k+j} + (m-1)M^*_{(i-1)k+p}/4 \le \xi_i \le \\ \sum_{j=1}^{p-1} M^*_{(i-1)k+j} + mM^*_{(i-1)k+p}/4-1, \text{ for } p=2,...,k. \end{cases} \quad (35)$$

Rule 4). Within each area in some ring, order of the points is defined with increase of the *u* coordinate. This means that within each area in some ring, the smallest index is assigned to the point with the smallest *u* coordinate and the highest index is assigned to the point with the highest *u* coordinate.

Now we will consider assignment of index ξ_i in the nonuniform part. Index ξ_i can take values $0 \le \xi_i \le N_2^*-1$. In the nonuniform region, rules 1), 3) and 4) are the same as for the uniform part. Rule 2) should be slightly modified, and it will be denoted as Rule 2').

Rule 2'). Rings are coded in increasing order, i.e. ring L_1+1 is coded firstly, after that ring L_1+2,…, and finally ring L_1+L_2. The highest four indexes ($\xi_i = N_2^*-4, N_2^*-3, \ N_2^*-2, N_2^*-1$) are assigned to the four overload representation points.

For the nonuniform part, rules 2') and 3) also can be joined in one rule:

Rule 2'-3). Indexes which are assigned to the points in the $(S+1)$-th region, in the L_1+p ring, in the *m*-th area ($p=1,...,L_2$; $m=1,2,3,4$) should satisfy the following condition:

$$\begin{cases} (m-1)M^*_{L_1+1}/4 \le \xi_i \le \\ mM^*_{L_1+1}/4-1, \text{ for } p=1; \\ \sum_{j=1}^{p-1} M^*_{L_1+j} + (m-1)M^*_{L_1+p}/4 \le \xi_i \le \\ \sum_{j=1}^{p-1} M^*_{L_1+j} + mM^*_{L_1+p}/4-1, \text{ for } p=2,...,L_2. \end{cases} \quad (36)$$

Decoding process in the receiver. There is a decoder for lossless code in receiver. This decoder receives codewords, and it should decide which representation points were coded with these codewords. The decoder has to know the following parameters: L_1, L_2, k, Q_i^* $(i=1,...,S+1)$, M_i^* $(i=1,...,L_1+L_2)$. This requires some memory space, but this memory space is very small (1kbit is enough in the most cases, which is not a problem in today electronic systems). Since $Q_i^* = 2^{\alpha_i}$, we can store α_i instead of Q_i^*, to save the memory space. Decoder uses Rules 1), 2-3), 2'-3) and 4) in the decoding process.

Decoding process is very simple. Decoder receives bit-stream and it should start with decoding process, i.e. it should decode the first codeword. Suppose that decoder knows which bit is the beginning of the first codeword. But, decoder does not know the length of the codeword (since it is variable). Decoder starts decoding of the codeword by counting consecutive 1's on the beginning of the codeword, until the first 0. This number of consecutive 1's is denoted with *z*, and decoder knows (using Rule 1) that $z=i-1$. It follows that the region *i* where the point belongs is $i=z+1$. Knowing *i*, decoder reads from its memory parameter Q_i^* and calculates the length of the codeword as $l_i = i+\log_2 Q_i^*$. Now, decoder knows where the end of the codeword is and where the next codeword begins. The last $\log_2 Q_i^*$ bits of the codeword is the natural binary code of the index ξ_i, and converting this natural binary code into integer, the index ξ_i is obtained. Knowing the index ξ_i and reading parameters M_i^* $(i=1,...,L_1+L_2)$ from its memory, decoder can apply Rule 2-3) if $1 \le i \le S$ or Rule 2'-3) if $i=S+1$. In that way, the ring and the area where the point belongs are determined. Then, using Rule 4) decoder can determine which point within this area was coded with this codeword. Decoding of the first codeword is finished and decoder continues to decode the next codewords in the same way.

2.7. Bit-rate

The bit rate R is defined as

$$R=\frac{1}{2}\sum_{i=1}^{S+1} l_i P_i \text{ [bits/sample].} \qquad (37)$$

The factor ½ means that R is bit-rate per dimension. $l_i = i + \log_2 Q_i^*$ is the length of the codewords in the i-th region. P_i is the probability of the i-th region, and it is defined with the following expression

$$P_i=\begin{cases} 4\int\limits_{r=(i-1)r^*/S}^{ir^*/S}\int\limits_{u=-r}^{r} f(r,u)dudr\,,\ 1\le i\le S\,; \\ 4\int\limits_{r=r^*}^{\infty}\int\limits_{u=-r}^{r} f(r,u)dudr\,,\ i=S+1\,. \end{cases} \qquad (38)$$

For $f(r,u)$ defined with (4), expression (38) becomes

$$P_i=\begin{cases} e^{-2(i-1)r^*/S}\left(1+2(i-1)r^*/S\right) \\ \qquad -e^{-2ir^*/S}\left(1+2ir^*/S\right), 1\le i\le S\,; \\ e^{-2r^*}\left(1+2r^*\right), i=S+1\,. \end{cases} \qquad (39)$$

3. Numerical results

In this section some numerical results for our two-dimensional quantizer will be given. We will present two examples.

Example 1. Let's consider two-dimensional quantizer with parameters: $N_1 = 9216$, $N_2 = 2^{11}$, $L_1 = 64$, $L_2 = 16$, $k = 8$. The optimal value for r^* is chosen to optimize performances. For this example, the optimal value is $r^* = 4.5$. Using expressions (22) and (37), it is obtained that $SNR = 34.2462$ dB and $R = 6.08946$ bits/sample. Entropy is $H = 6.04276$ bits/sample (entropy is calculated using well-known expression $\sum_i p_i \log_2(1/p_i)$ where p_i represents point's probability, and summation is done over all points). This quantizer is designed to satisfy the G.712 standard. The modified number of points in the uniform part is $N_1^* = 8444$. Since N_2 is a power of two, there is no need to modify the number of points in the nonuniform part, i.e. $N_2^* = N_2$.

Simulation in MATLAB for this example is done for memoryless Laplacian source. Results of the simulation are: $SNR = 34.2447$dB and $R = 6.0435$ bits/sample. We can see that simulation and theoretical results are very close (almost identical). In that way, correctness of the theory developed in Section 2 is proven.

The average number of points which is used during the quantizer's work is $\overline{N} = (1-P_{S+1})N_1^* + P_{S+1}N_2^*$ (explanation – cells in the uniform and in the nonuniform part are not used simultaneously; instead of that, in one moment only cells in the nonuniform part are used with probability P_{S+1} given with (39), or only cells in the uniform part are used with the probability $1-P_{S+1}$). For the aim of comparison with scalar quantizers, we can define equivalent scalar quantizer of our vector quantizer, whose the average number of points used during quantizer's work is $N_{av} = \sqrt{\overline{N}}$ (measured in points per dimension). N_{av} shows the execution complexity (complexity of execution of coding and decoding algorithm), which is very important for the speed of the execution. For this example it is obtain that $N_{av} = 92$ points. So, our model can satisfy the G.712 standard with 92 points per dimension in average. It is known [1] that scalar uniform quantizer can satisfy the G.712 standard using 256 points and scalar nonuniform quantizer using 128 points during the process of execution. So, we can conclude that our model can satisfy the G.712 standard using much smaller number of points per dimension during the execution process, compared to scalar uniform and nonuniform quantizers. Therefore, the time that is needed for coding and decoding process is shorter.

Example 2. In this example, we will consider the two-dimensional quantizer with the following parameters: $N_1 = 256$, $N_2 = 64$, $L_1 = 8$, $L_2 = 4$, $k = 2$. For these parameters, the optimal value for r^* is $r^* = 2.8$. It is obtained that $SNR = 19.2187$ dB, $R = 3.7586$ bits/sample and $H = 3.678$ bits/sample. Since this quantizer has small number of rings, the design process will be described in more details. In the uniform part we have 8 rings. Using expression (27), we calculate numbers of points in these rings: $M_i = \{11, 26, 35, 39, 39, 38, 36, 32\}$. Now, we join two successive rings ($k = 2$) into one region with $Q_i = \{37, 74, 77, 68\}$ points. Now, we modify numbers of points in regions to obtain the power of two: $Q_i^* = \{32, 64, 64, 64\}$. The modified numbers of points in rings are: $M_i^* = \{8, 24, 28, 36, 32, 32, 32, 32\}$. The modified number of points in the uniform part is $N_1^* = 224$. In the nonuniform part we have four rings with $M_i = \{20, 16, 16, 8\}$ points and four overload points. Since N_2 is a power of two, there is no need to modify the number of points in the nonuniform part. Here it is valid that $N_{av} = 15$ points per dimension. It is known [1] that scalar uniform quantizer with 16 points (i.e. bit-rate of 4 bits/sample) gives

$SNR = 15.96$ dB and scalar nonuniform quantizer with 16 points gives $SNR = 18.13$ dB. So, our model gives much better performances using smaller number of points per dimension, compared to scalar uniform and nonuniform quantizers.

From these examples we can also conclude that the lossless code is very efficient, since the bit-rate *R* is very close to the entropy *H*, i.e. the lossless code is very close to the ideal code.

4. Discussion

In this section we will give some additional explanations and remarks about our model.

1) Why a hybrid quantizer is used for *r* coordinate quantization? As it was said, hybrid quantizer is a combination of uniform and nonuniform quantizers. Both of them have some good characteristics. Uniform quantizer is very suitable to use together with lossless code, giving small bit-rate. Nonuniform quantizer gives higher *SNR*. Hybrid quantizer joins good characteristics of both uniform and nonuniform quantizers. Now, two examples will be given (the first where only uniform quantizer is used for *r* coordinate and the second where only nonuniform quantizer is used for *r* coordinate), and comparison with Example 2 (where hybrid quantizer is used for *r* coordinate) will be done.

Example 3. Two-dimensional quantizer where only uniform quantizer is used for *r* coordinate, with $N_1 = 256$ points and $L_1 = 8$ rings. In this case optimal range of the uniform quantizer is $r^* = 6$. Since there is no nonuniform quantizer in the outer part, when $r > r^*$ we have overload distortion. It is obtained that $SNR = 14.2018$ dB and $R = 3.6991$ bits/sample. These results can be compared with results of Example 2 (where hybrid quantizer for *r* coordinate was used) We can see that the bit-rate is almost the same, but *SNR* is much higher (for 5 dB) when hybrid quantizer for *r* coordinate is used. Therefore, hybrid quantizer for *r* coordinate is better than uniform quantizer.

Example 4. Two-dimensional quantizer with 256 points and 16 rings, where only nonuniform quantizer is used for *r* coordinate. Performances are: $SNR = 19.1085$ dB and $R = 4.222$ bits/sample. We can see (from Example 2) that when hybrid quantizer is used for *r* coordinate, almost the same *SNR* is achieved with smaller (for 0.47 bits/sample) bit-rate *R*. Therefore, hybrid quantizer for *r* coordinate is better than nonuniform quantizer.

So, we can conclude that hybrid quantizer is the best solution for quantization of *r* coordinate.

2) Execution complexity. It was shown in Section 3 that our model can achieve the same performances using smaller number of points per dimension, compared to scalar uniform and nonuniform quantizers. This means that our model has low execution complexity. Now, we will further analyze execution complexity of the model, which is equal to the sum of the execution complexity of the two-dimensional quantizer and of the lossless coder.

2a) Execution complexity of the two-dimensional quantizer. It is well known that execution complexity of nonuniform quantizer is much higher than that of the uniform quantizer. As it was said, *u* coordinate is quantized with uniform quantizer, which is simple. *r* coordinate is quantized with hybrid quantizer, which is the combination of uniform quantizer (which is simple) and nonuniform quantizer. This nonuniform quantizer has small number of levels (equal to L_2). Furthermore, this nonuniform quantizer is very rarely used, since it is placed in the outer part of the two-dimensional plane (for high *r*), which has very small probability. The great majority of input vectors (*r*, *u*) belong to the uniform part, and then two-dimensional quantizer can be realized as combination of two uniform quantizers (one for *r* and one for *u* coordinate), which is simple. So, we can conclude that nonuniform quantizer for *r* coordinate does not increase execution complexity of two-dimensional quantizer significantly.

To better explain this, the quantizer from Example 1 (from Section 3) will be considered. *r* coordinate is quantized using uniform quantizer with $L_1 = 64$ levels and nonuniform quantizer with $L_2 = 16$ levels. Probability of the nonuniform part is $P_{S+1} = 0.0012341$. Now, we will determine how much levels are needed for the uniform quantizer for *u* coordinate. When we know the ring and the area where some point belongs, uniform quantization for *u* coordinate should be done only within this area in this ring. So, if the point belongs to the *i*-th ring, the number of levels of the uniform quantizer is equal to the optimal number of points in that area, which is equal to $M_i^* / 4$, and it varies from one ring to another. For this example, the maximal value of $M_i^* / 4$ is 65. Therefore, the uniform quantizer for *u* coordinate should have maximum 65 levels, but for the most rings this number is smaller. Finally, we can conclude that: in 99.877 % cases, the two-dimensional quantizer is a combination of two uniform quantizers (one with 64 levels for *r* coordinate and one with maximum 65 levels for *u* coordinate); in 0.123 % cases the two-dimensional quantizer is a combination of one nonuniform quantizer with 16 levels (for *r* coordinate) and one uniform quantizer with maximum 65 levels (for *u* coordinate). Therefore, the execution complexity of this two-dimensional quantizer is smaller compared to scalar uniform quantizer with 256 levels or scalar nonuniform quantizer with 128 levels, which can achieve the same quality (defined with the G.712 standard).

2b) Execution complexity of the lossless code. The most often used lossless code is Huffman code. But, the complexity of Huffman code is very high, since code tree should be made during the coding process (in transmitter) as well as during the decoding process (in receiver). Because of that, Huffman code is suitable only for discrete sources with small number of symbols. Huffman code can be used for coding of output levels of scalar quantizer if the number of levels is not too high. But, the number of representation points for two-dimensional quantizers can be very large. Huffman code is not suitable to be used in combination with two-dimensional quantizers.

Lossless code described in this paper is very simple and it can be very easily applied for coding points of two-dimensional quantizers. For this code there is no need to make a code tree. Coding and decoding processes are very simple – they include counting of 1's and conversion of index ξ_i into natural binary code (and reverse). This code is a modification of Golomb-Rice code. It is well known that Golomb-Rice code is much simpler than Huffman code [11].

Huffman code also needs knowledge about probabilities of every point. Since we have large number of points in two-dimensional quantizer, calculation of these probabilities is very difficult. Our lossless code does not require knowledge of points' probabilities. This is another reason why our lossless code is much simpler than Huffman code.

It was shown in Section 3 that our lossless code, although very simple, is very efficient, since it gives the bit-rate *R* very close to the entropy *H*.

We proved that both the two-dimensional quantizer and the lossless coder have low execution complexity, therefore the total execution complexity of our model is low.

5. Conclusion

In this paper, one solution was given for the problem which had not been considered in literature: application of the lossless code on output levels of the two-dimensional quantizer. Huffman code is inapplicable for this problem because it is too complex. Design of the two-dimensional quantizer was done using Helmert transform, and optimization of the number of cells on rings was done. New lossless code was introduced and applied on the two-dimensional quantizer. It is very simple, but it was shown that it is very close to the ideal code, since the bit-rate is very close to the entropy. Simulation in MATLAB was done and theoretic results were proved. It was shown that our two-dimensional model had smaller execution complexity than scalar uniform and nonuniform quantizers, i.e. it could achieve the same performances using much smaller number of points per dimension and faster.

References

[1] **N.S. Jayant, P. Noll.** Digital Coding of Waveforms. *Prentice-Hall*, 1984.

[2] **Z. Peric, J. Nikolic, A. Mosic, S. Panic.** A switched-adaptive quantization technique using µ-law quantizers. *Information Technology and Control*, 39(4), 2010, 317-320.

[3] **Z. Peric, J. Nikolic, Z. Eskic, S. Krstic, N. Markovic**. Design of novel scalar quantizer model for Gaussian source. *Information Technology and Control*, 37(4), 2008, 321-325.

[4] **A. Gersho, R.M. Gray.** Vector Quantization and Signal Compression. *Kluwer Academic Publishers, Masachusetts*, 1992.

[5] **D. Hankerson, G.A. Harris, P.D. Jr. Johnson.** Introduction to information theory and data compression. *CHAPMAN & HALL/CRC*, 2*nd edition*, 2004.

[6] **K. Sayood.** Introduction to Data Compression. *Elsevier Inc*., 3*rd edition*, 2006.

[7] **A.R. Elabdalla, M.Irshid.** An efficient bitwise Huffman coding technique based on source mapping. *Computers and Electrical Engineering*, 27, 2001, 265-272.

[8] **K. Fredriksson, J. Tarhio.** Efficient String Matching in Huffman Compressed Texts. *Fundamenta Informaticae*, 63(1), 2004, 1-16.

[9] **A. Zolghadr-E-Asli, S. Alipour.** An effective method for still image compression /decompression for transmission on PSTN lines based on modifications of Huffman coding. *Computers and Electrical Engineering*, 30, 2004, 129-145.

[10] **Z. Peric, M. Petkovic, M. Dincic.** Simple Compression Algorithm for Memoryless Laplacian Source Based on the Optimal Companding Technique. *Informatica*, 20(1), 2009, 99-114.

[11] **D. Salomon.** Variable-length Codes for Data Compression. *Springer*, 2007.

[12] **R.F. Rice, P.S. Yeh, W.H. Miller.** Algorithms for high speed universal noiseless coding module. *Proc. of the AIAA Computing in Aerospace Conference, San Diego*, 1993.

[13] **H. Malvar.** Adaptive Run-Length/Golomb-Rice Encoding of Quantized Generalized Gaussian Sources with Unknown Statistics. *Proc. of the Data Compression Conference (DCC), IEEE Computer Society*, 2006, 23-32.

[14] **M. Weinberger, G. Seroussi, G. Sapiro.** The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Trans. Image Processing*, 9, 2000, 1309–1324.

[15] **T. Liebchen, Y. Reznik.** MPEG-4 ALS: an Emerging Standard for lossless Audio Coding. *Proc. of the Data Compression Conference (DCC), IEEE Computer Society*, 2004, 439-448.

[16] **Consultative Committee for Space Data Systems (CCSDS).** Lossless Data Compression. *Blue Book, Issue 1*, 1997.

[17] **P.C. Tsai, S.J. Wang, C.H. Lin, T.H. Yeh.** Test Data Compression for Minimum Test Application Time. *Journal of Information Science and Engineering*, 23, 2007, 1901-1909.

[18] **W.D. Leon-Salas, S. Balkir, K. Sayood, W.M. Hoffman.** An Analog-to-Digital Converter with Golomb-Rice Output Codes. *IEEE Transactions on Circuits and Systems-II: Express Briefs*, 53, 2006, 278-282.

[19] **Z. Perić**, **M. Dinčić**, **M. Petković**. Design of a hybrid quantizer with variable length code. *Fundamenta Informaticae* 98(2-3), 2010, 233-256.

[20] **Z. Peric, M. Novkovic, V. Despotovic.** Linearization Method for Two-dimensional Memoryless Laplass Source. *Electronics and Electrical Engineering,* 73(1), 2007, 41-44.

Received July 2010.