Design of Optimal Entropy-constrained Unrestricted Polar Quantizer for Bivariate Circularly Symmetric Sources

Huihui Wu, *Student Member, IEEE* and Sorina Dumitrescu, *Senior Member, IEEE*

***Abstract*—This paper proposes an algorithm for the design of entropy-constrained unrestricted polar quantizer (ECUPQ) for bivariate circularly symmetric sources. The algorithm is globally optimal for the class of ECUPQs with magnitude quantizer thresholds confined to a finite set.**

The optimization problem is formulated as the minimization of a weighted sum of distortion and entropy and the proposed solution is based on modeling the problem as a minimum-weight path problem in a certain weighted directed acyclic graph. Each graph edge corresponds to a possible magnitude quantizer bin and computing its weight involves solving another optimization problem. We develop a fast strategy for evaluating all edge weights, leading to a $O(K^2 + KP_{max})$ time solution algorithm, where K is the size of the set of possible magnitude thresholds and P_{max} is the maximum number of phase levels.

The practical performance of the proposed algorithm is assessed for a bivariate circularly symmetric Gaussian source, at rates ranging from 0.5 to 6.0 bits/sample. Our results demonstrate that the proposed approach achieves performance very close to the asymptotically optimal ECUPQ at all rates, while at low rates it significantly outperforms all previous UPQ schemes. Notably, peak improvement of 0.755 dB can be achieved for rates below 2.5.

***Index Terms*—Unrestricted polar quantization, entropy-constrained quantizer, globally optimal algorithm, minimum weight path problem.**

I. Introduction

A polar quantizer quantizes the magnitude and the phase of a two dimensional source vector represented in polar coordinates. The phase quantizer is uniform while the magnitude quantizer may be nonuniform. Polar quantization of bivariate sources with circularly symmetric densities, has been extensively investigated either for the general case or for the specific Gaussian case [1]- [19].

Early work on polar quantization uses independent quantizers for the two components. Such a scheme is referred in the literature as a *strictly* polar quantizer (SPQ) [4] or a *conventional* polar quantizer [11]. In later work the *unrestricted* polar quantizer is introduced (UPQ) [7], where the phase quantizer depends on the magnitude level, which is shown to outperform SPQ.

Polar quantization is useful in numerous applications, such as image processing [1], [20], for the encoding of discrete Fourier transform coefficients [2], [3], in holographic image processing [21], as well as for the quantization of sinusoid signals with application in audio coding [14]. More recently, polar quantization was also used for wireless receiver design in [22].

Most of the work on the analysis and design of polar quantizers relies on the high resolution assumption. In particular, the asymptotic analysis of the uniform polar quantizers, i.e., where the quantizer of the magnitude is also uniform, was performed in [9], [12], [17] for the conventional case and in [13] for the unrestricted case. We point out that the above mentioned papers assume level-constrained (or fixed-rate) quantization, i.e., where the goal is to minimize the distortion for a fixed number of total quantization levels. The asymptotic analysis of level-constrained non-uniform UPQ was addressed in [10], [11], [16].

The design of optimal practical polar quantizers, i.e., without the high rate assumption, was considered in [2], [4] for the level-constrained SPQ and in [7] for the level-constrained UPQ. The approach taken in the aforementioned work is to solve iteratively the necessary conditions for optimal decision thresholds and optimal reconstruction values. This iterative procedure can be applied when the number M of magnitude levels and the number P of phase levels are fixed, in the case of SPQ, respectively, when the M-tuple of numbers of phase levels $(P_1, \cdots, P_M)$ is fixed, in the case of UPQ. More specifically, since each phase quantizer is uniform the problem further reduces to finding the optimal decision thresholds and reconstruction levels of the magnitude quantizer, which depend on the number of phase levels of the phase quantizer(s). The latter problem is solved in [4], [7] by using an iterative algorithm similar to Max-Lloyd algorithm [23], [24] for optimal scalar quantizer design, i.e., by iteratively optimizing the encoder, respectively the decoder, while the other component is kept fixed. However, the aforementioned works do not find an efficient solution for optimizing the rate allocation between the magnitude and phase quantizers, i.e. for finding the optimum pair (M, P) satisfying the constraint $MP = N$ in the case of SPQ, respectively, finding the optimum configuration $(M, P_1, \cdots, P_M)$ satisfying $\sum_{m=1}^{M} P_m = N$, where N is the total number of polar quantizer bins. In absence of an efficient strategy the authors of [4], [7] rely on exhaustive search to optimize the rate allocation.

To increase the efficiency of the polar quantizer, entropy coding may be applied to the quantizer's outputs. This was done, for instance, in [7]. However, for optimal performance the polar quantizer has to be optimized under a constraint on the entropy. Such a quantizer is called entropy-constrained

The authors are with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada (Emails: wuh58@mcmaster.ca;sorina@mail.ece.mcmaster.ca).

quantizer. Work [14] is the only work addressing the design of entropy-constrained polar quantizers, up to our knowledge. The authors of [14] derive the asymptotically optimal entropy-constrained UPQ (ECUPQ) and entropy-constrained SPQ (ECSPQ), as the rate approaches infinity. They further consider a bivariate circularly symmetric Gaussian source and compare the performance of the proposed ECUPQ to other asymptotically optimal entropy-constrained quantizers. As expected, they find that the asymptotical performance of ECUPQ is significantly superior to that of ECSPQ. They also perform the comparison against the entropy-constrained rectangular quantizer (ECRQ), which uses scalar quantization of each Cartesian coordinate. This comparison reveals that the performance of ECUPQ and ECRQ are identical asymptotically. This conclusion is expected since, as the rate approaches ∞, the shape of most of the UPQ quantizer cells approaches a rectangular shape. Moreover, the authors of [14] show that the practical performance of the proposed ECUPQ is close to the performance predicted by the asymptotic expression, when the rates are high enough.

As the results of [14] illustrate, the asymptotical expression of ECUPQ performance is not accurate if the rate is not sufficiently high. In particular, in our implementation of the ECUPQ proposed in [14] we found that the gap to the asymptotic performance is higher than 0.5 dB for rates between 2.050 and 2.495 bits per sample, and, although the gap gradually decreases, it remains higher than 0.1 dB, for rates up to 4.0 bits/sample. Additionally, the asymptotic expression cannot be applied to rates smaller than $0.5\log_2(2\pi e) \approx 2.047$, thus no comparison is possible for those rates. Furthermore, the optimality of the ECUPQ of [14] holds as the rate approaches infinity, but it is not guaranteed at finite rates.

The above observations raise two natural questions that motivate our work:

Q1) Is it possible to further improve the performance of ECUPQ at finite rates?
Q2) Does ECUPQ exhibit any advantage in terms of performance versus ECRQ at finite rates?

In order to address these inquiries we propose the design of ECUPQ for a bivariate circularly symmetric source, at finite rates. Our design is globally optimal for the class of ECUPQs with thresholds of the magnitude quantizer restricted to some finite set. In practice this finite set can be a fine discretization of the interval $[0, B]$ for some sufficiently large B.

We formulate the problem of optimal ECUPQ design as the minimization of the Lagrangian for a given multiplier λ, which is the same formulation as in [14]. Thus, the cost function is actually a weighted sum of the quantizer distortion and entropy. This formulation readily simplifies the problem of rate allocation between the magnitude quantizer and phase quantizers. Specifically, for each bin of the magnitude quantizer the optimal number of phase levels of the phase quantizer can be determined independently of other bins. This observation is critical for our approach since it allows us to convert the cost function (after determining the optimal phase quantizer corresponding to each magnitude bin) to a summation of the costs of individual magnitude bins. Thus, this problem can be modeled as a minimum-weight path (MWP) problem in a certain weighted directed acyclic graph (WDAG), where each edge represents a possible bin of the magnitude quantizer. In order to expedite the computation of all weights we develop a fast strategy for finding the optimal number of phase levels for all possible magnitude bins. The overall running time of the solution algorithm is $O(K^2 + KP_{max})$ where K is the size of the set from which the magnitude thresholds are selected, while P_{max} is an upper bound for the number of phase levels corresponding to a magnitude bin.

Enabled with this tool we proceed to answer the initial questions Q1 and Q2. For this we have tested the proposed ECUPQ design algorithm for a bivariate Gaussian source for rates up to 6 bits/sample. Our experiments show that the proposed approach outperforms both the entropy-coded UPQ of [7] and the practical ECUPQ of [14] designed based on the high rate assumption. The comparison against the UPQ of [7] was performed using the results reported in [7], which cover a rate range between 0.5 and 2.5 bits/sample. The gain of our scheme over the scheme of [7] ranges from 0.216 to 0.755 dB, and is always higher than 0.6 dB when the rate is larger than 1.5 bits/sample. On the other hand, the comparison against the practical ECUPQ of [14] was carried out for rates higher than 2.050. The improvement over the latter scheme is higher than 0.5 dB for rates in the range 2.050 to 2.495 bits/sample and remains higher than 0.1 dB for rates up to 4.0 bits/sample. Additionally, we have observed that the performance of our design is very close to the asymptotic ECUPQ performance derived in [14]. We have also compared the proposed ECUPQ with the ECRQ obtained using the algorithm of [28] for optimal entropy-constrained scalar quantizer design. We found that ECUPQ has an advantage in terms of performance versus ECRQ, even if small, for rates between 0.5 and 2.256. Notably, the highest improvements are achieved for rates ranging from 1.0 to 1.377 and reach values higher than 0.1 dB. In conclusion, our results show that the benefit of the proposed ECUPQ scheme is most prominent for rates up to about 2.5 bits/sample. It is important to emphasize that this range of encoding rates is of interest in lossy image coding, especially for applications such as network image transmission or remote sensing. Actually, one of the reasons of the development of the JPEG2000 image compression standard was the need to improve the performance at low bit rates [25].

We point out that the design approach based on modeling the problem as an MWP problem in some WDAG, with or without a constraint on the number of edges, has been used in the past for the design of other scalar quantizer systems. For instance, it was employed for the design of fixed-rate quantizers [26], entropy-constrained quantizers [27], [28], Wyner-Ziv quantizers [27], [28], multi-resolution and multiple description quantizers [27]–[31], joint source-channel quantizer with random index assignment [32], as well as quantizers for sequential source coding [33]. Other works using dynamic programming to solve scalar quantization problems [34] also implicitly solve an MWP problem, even if they do not formulate it as such. The aspect which distinguishes the most our work from the aforementioned work, is that the optimization problem that needs to be solved in order to compute the weight of a graph edge is of a different nature. Another notable contribution of our work is the efficient handling of all these optimization problems, fact which allows for the computation of each edge weight to be performed in constant time, on average, when $P_{max} = O(K)$.

In summary, the main contribution of our work is as follows.

a) We propose the first globally optimal ECUPQ design algorithm for finite rates. The optimality claim holds for the class of UPQs with the thresholds for the magnitude quantizer restricted to a finite set.
b) Practical results for a bivariate memoryless Gaussian source show that for rates up to 2.5 bits/sample our algorithm considerably outperforms the best entropy-coded and entropy-constrained UPQ schemes known to date.
c) Our proposed algorithm is the first algorithm for practical polar quantizer design, which handles efficiently the problem of rate allocation between the magnitude and phase quantizers.

We would like to point out that [35] is a shortened conference version of this work. In [35] the computation of each edge weight is handled using binary search, which leads to a running time of $O(K^2 \log_2 P_{max})$ operations for the overall design algorithm. In the present work we propose a faster method to compute the edge weights when $\frac{P_{max}}{\log_2 P_{max}} < K$, leading to the total time complexity of $O(K^2 + KP_{max})$. Paper [35] does not include Proposition 2 and the proof of Proposition 1. Additionally, the current work contains a more thorough experimental performance evaluation of the proposed approach than [35]. Specifically, more rates are examined, the configuration of the proposed ECUPQ at these rates is included and discussed, the comparison with ECRQ and with the asymptotical performance of the two-dimensional entropy-constrained vector quantizer, are added.

The rest of the paper is organized as follows. The next section introduces the notations and formulates the optimization problem. Section III shows how the problem can be modeled as the MWP in some WDAG. Section IV proposes an efficient strategy for computing the optimal number of phase levels for all possible magnitude bins and finalizes the design algorithm. The experimental results and their discussion are presented in Section V, while Section VI concludes the paper.

II. Problem Formulation

Consider a bivariate random variable with the following circularly symmetric density, as a function of the polar coordinates r and θ,

$$p(r,\theta) = \frac{1}{2\pi} g(r),\ 0 \le r < \infty,\ 0 \le \theta < 2\pi.$$

Note that $g(r)$ is the marginal probability density function (pdf) of the magnitude variable, while the phase variable is uniformly distributed over the interval $[0, 2\pi)$. Additionally, notice that the magnitude and phase variables are independent. An example of such a variable is a two-dimensional memoryless Gaussian vector (X_1, X_2), i.e., where X_1 and X_2 are independent and have identical marginal pdfs. Quantization of Gaussian variables is interesting since it has numerous practical applications. For example, the joint distribution of discrete Fourier transform coefficients of a stationary data sequence is asymptotically Gaussian [3]. Also, the probability density function of prediction error signal in a differential pulse code modulation coder for moving pictures can be modeled as Gaussian [36].

Let M denote the number of magnitude levels of the UPQ and let $\mathbf{r} \triangleq (r_1, r_2, \cdots, r_{M-1})$ denote the vector of thresholds of the magnitude quantizer, where

$$r_0 = 0 < r_1 < r_2 < \cdots < r_{M-1} < r_M = \infty.$$

For $1 \le m \le M$, let C_m denote the m-th cell (or bin) of the magnitude quantizer, i.e., $C_m = \{r | r_{m-1} \le r < r_m\}$. Further, let $\mathbf{P} \triangleq (P_1, P_2, \cdots, P_M)$, where P_m denotes the number of phase regions of the phase quantizer corresponding to C_m, $1 \le m \le M$. Each phase quantizer is uniform, consequently, each quantization bin of the UPQ can be represented as

$$\mathcal{R}(m,s) = \left\{ re^{j\theta} | r_{m-1} \le r < r_m, (s-1)\frac{2\pi}{P_m} \le \theta < s\frac{2\pi}{P_m} \right\},$$

for $1 \le m \le M$, and $1 \le s \le P_m$. Clearly, the total number of quantization bins of the UPQ is $N = \sum_{m=1}^{M} P_m$.

The reconstruction for quantizer bin $\mathcal{R}(m,s)$ is $A_m e^{j\theta_{m,s}}$, where A_m is the reconstruction value of the magnitude for the m-th magnitude level, and $\theta_{m,s}$ is the reconstruction value for the phase.

We will use the squared error as a distortion measure. Therefore, the expected distortion (per sample) of the UPQ can be expressed as [1], [2], [7]

$$\begin{aligned} D &= \frac{1}{2} \sum_{m=1}^{M} \sum_{s=1}^{P_m} \int_{r_{m-1}}^{r_m} \int_{(s-1)\frac{2\pi}{P_m}}^{s\frac{2\pi}{P_m}} \| re^{j\theta} - A_m e^{j\theta_{m,s}} \|^2 p(r,\theta) d\theta dr \\ &= \frac{1}{2} \sum_{m=1}^{M} \sum_{s=1}^{P_m} \int_{r_{m-1}}^{r_m} \int_{(s-1)\frac{2\pi}{P_m}}^{s\frac{2\pi}{P_m}} (r^2 + A_m^2 - \\ &\qquad 2rA_m \cos(\theta - \theta_{m,s})) \frac{g(r)}{2\pi} d\theta dr. \end{aligned} \tag{1}$$

The best reconstruction values, which minimize the distortion, were determined in prior work [1], [2], [7] by solving $\partial D / \partial \theta_{m,s} = 0$ and $\partial D / \partial A_m = 0$, leading to

$$\theta_{m,s} = (2s-1)\pi / P_m, \tag{2}$$

$$A_m = sinc\left(\frac{1}{P_m}\right) \frac{\int_{r_{m-1}}^{r_m} r g(r) dr}{\int_{r_{m-1}}^{r_m} g(r) dr}, \tag{3}$$

where $sinc(\frac{1}{P_m}) = \frac{\sin(\pi/P_m)}{\pi/P_m}$. By exploiting (2) and (3), the expected distortion can be simplified as

$$\begin{aligned} D &= \frac{1}{2} \left(\sum_{m=1}^{M} \int_{r_{m-1}}^{r_m} r^2 g(r) dr - \sum_{m=1}^{M} A_m^2 \int_{r_{m-1}}^{r_m} g(r) dr \right) \\ &= \frac{1}{2} \left(\int_0^\infty r^2 g(r) dr - \sum_{m=1}^{M} A_m^2 \int_{r_{m-1}}^{r_m} g(r) dr \right). \end{aligned} \tag{4}$$

Notice that, since the reconstruction values of the UPQ are given by (2) and (3), it follows that the tuples $\mathbf{r}$ and $\mathbf{P}$ completely specify the UPQ.

Let I_a and I_θ denote the random variables representing the magnitude and phase quantization indexes, respectively. Let

$H(I_a, I_\theta)$ denote the joint entropy of (I_a, I_θ), which can be expressed as follows

$$\begin{aligned} H(I_a, I_\theta) &= H(I_a) + H(I_\theta|I_a) \\ &= \sum_{m=1}^{M} q(m)(-\log_2 q(m) + \log_2 P_m), \end{aligned} \quad (5)$$

where $q(m)$ is the probability of m-th magnitude level, i.e., $q(m) = \int_{r_{m-1}}^{r_m} g(r)dr$. Then the entropy of the UPQ (in bits/sample) is defined $H(I_a, I_\theta)/2$.

Following prior work on entropy-constrained quantization [14], [28], [37] we formulate the problem of ECUPQ design as follows

$$\min_{M,\mathbf{r},\mathbf{P}} \quad \mathcal{L}(\mathbf{r}, \mathbf{P}, \lambda), \quad (6)$$

for fixed Lagrangian multiplier $\lambda > 0$, where

$$\mathcal{L}(\mathbf{r}, \mathbf{P}, \lambda) \triangleq D + \lambda H(I_a, I_\theta)/2.$$

It is known [38], [39] that the set of solutions to problem (6), when λ varies over $(0, \infty)$, is the set of UPQs such that the corresponding pair $(H(I_a, I_\theta)/2, D)$ is on the lower boundary of the convex hull of the set of all possible pairs $(H(I_a, I_\theta)/2, D)$. Thus, a UPQ which is a solution to problem (6) minimizes the distortion for the corresponding entropy, thus it is an ECUPQ.

We will solve problem (6) under the constraint that the thresholds of the UPQ take values in a finite set $\mathcal{A} = \{a_1, a_2, \cdots, a_K\}$. This set can be obtained by finely discretizing the interval $[0, B]$, for some B chosen such that the probability that the magnitude level is larger than B, to be sufficiently small.

Thus, the problem that we will solve in this work is the following

$$\begin{aligned} &\min\nolimits_{M,\mathbf{r},\mathbf{P}} \quad \mathcal{L}(\mathbf{r}, \mathbf{P}, \lambda), \\ &\text{subject to } r_i \in \mathcal{A}, 1 \le i \le M-1. \end{aligned} \quad (7)$$

Its solution is discussed in the following two sections.

III. Graph Model

In this section we show how the minimization problem (7) can be modeled as an MWP problem in a certain WDAG. For this we need first to perform some manipulation of the cost function. Notice that the first term in (4) is constant, therefore we can remove it from the cost function. Thus, minimizing $\mathcal{L}(\mathbf{r}, \mathbf{P}, \lambda)$ is equivalent to minimizing $C(\mathbf{r}, \mathbf{P})$, where

$$C(\mathbf{r}, \mathbf{P}) \triangleq \frac{1}{2}\left(-\sum_{m=1}^{M} A_m^2 \int_{r_{m-1}}^{r_m} g(r)dr + \lambda H(I_a, I_\theta)\right).$$

Further, substituting (3) and (5) into the above equation leads to

$$C(\mathbf{r}, \mathbf{P}) = \frac{1}{2}\sum_{m=1}^{M} \int_{r_{m-1}}^{r_m} g(r)dr \left(-sinc^2\left(\frac{1}{P_m}\right) x_m^2 + \lambda \log_2 \frac{P_m}{\int_{r_{m-1}}^{r_m} g(r)dr}\right), \quad (8)$$

where $x_m = \frac{\int_{r_{m-1}}^{r_m} rg(r)dr}{\int_{r_{m-1}}^{r_m} g(r)dr}$.

Now it can be seen that if the vector of thresholds $\mathbf{r}$ is fixed, then P_m can be optimized separately for each m. Specifically, the optimal value of P_m, $1 \le m \le M$, is

$$P_m^* = \arg\min_{P_m}\left(-sinc^2\left(\frac{1}{P_m}\right) x_m^2 + \lambda \log_2 P_m\right),$$

since $\int_{r_{m-1}}^{r_m} g(r)dr$ and x_m are fixed, for fixed $\mathbf{r}$.

Consider now the following notations. For each $0 \le \alpha < \beta \le \infty$, denote

$$q(\alpha, \beta) \triangleq \int_\alpha^\beta g(r)dr,$$

$$x(\alpha, \beta) \triangleq \frac{\int_\alpha^\beta rg(r)dr}{\int_\alpha^\beta g(r)dr},$$

$$P^*_{(\alpha,\beta)} \triangleq \min \arg\min_P \left(-sinc^2\left(\frac{1}{P}\right) x(\alpha, \beta)^2 + \lambda \log_2 P\right), \quad (9)$$

where the minimization is over all positive integers P. Note that, if there are more values P minimizing the cost in (9), we select the smallest one as $P^*_{(\alpha,\beta)}$.

Further, by replacing P_m in (8) by $P^*_{(r_{m-1},r_m)}$, we obtain a new cost function which only depends on $\mathbf{r}$

$$\bar{C}(\mathbf{r}) \triangleq \frac{1}{2}\sum_{m=1}^{M} q(r_{m-1}, r_m)\left(\lambda \log_2 \frac{P^*_{(r_{m-1},r_m)}}{q(r_{m-1}, r_m)} - sinc^2\left(\frac{1}{P^*_{(r_{m-1},r_m)}}\right) x(r_{m-1}, r_m)^2\right). \quad (10)$$

According to the above discussion, problem (7) is equivalent to the following

$$\begin{aligned} &\min\nolimits_{M,\mathbf{r}} \quad \bar{C}(\mathbf{r}) \\ &\text{subject to } r_i \in \mathcal{A}, 1 \le i \le M-1. \end{aligned} \quad (11)$$

The next step is based on the observation that the cost $\bar{C}(\mathbf{r})$ can be expressed as a summation of costs of the individual intervals (r_{m-1}, r_m), fact which allows us to regard it as the weight of a path in a certain WDAG, as we show next.

Let us assume that the elements of $\mathcal{A}$ are labeled in increasing order, i.e., $0 < a_i < a_{i+1}$, for $1 \le i \le K-1$. Additionally, let us denote $a_0 = 0$ and $a_{K+1} = \infty$. Construct now the WDAG $G = (V, E, w)$, where $V = \{0, 1, 2, \cdots, K+1\}$ is the vertex set, and $E = \{(u, v) \in V^2 \mid 0 \le u < v \le K+1\}$ is the edge set. Further, the weight of each edge (u, v) is defined as follows,

$$w(u, v) \triangleq \frac{1}{2} q(a_u, a_v)\left(-sinc^2\left(\frac{1}{\hat{P}(u, v)}\right) x(a_u, a_v)^2 + \lambda \log_2 \frac{\hat{P}(u, v)}{q(a_u, a_v)}\right), \quad (12)$$

where we use the new notation $\hat{P}(u, v)$ instead of $P^*_{(a_u,a_v)}$, for simplicity.

The source node in this graph is vertex 0 and the final node is $K+1$. A path in this graph from some node u to some node

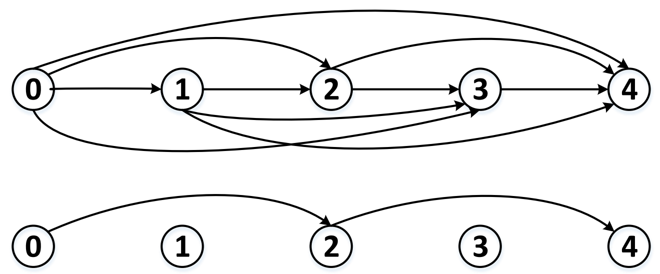


Fig. 1: Illustration of the graph G (top) for $K = 3$ and a path in the graph (bottom). Nodes are depicted with circles and edges with arcs. The path shown on the bottom corresponds to the magnitude quantizer with bins $[0, a_2)$ and $[a_2, \infty)$.

v is any sequence of connected edges starting at u and ending at v. Clearly, any path from the source to the final node can be represented as an $(s+1)$-tuple of vertexes $\mathbf{t} = (t_0, t_1, \cdots, t_s)$, satisfying $t_0 = 0$, $t_s = K+1$ and $t_{m-1} < t_m$, $1 \leq m \leq s$, for some $s \geq 1$. Note that s equals the number of edges on the path. Let us denote by $\mathcal{T}(s)$ the set of all paths from the source to the final node with exactly s edges, for each $s \geq 1$. The weight $W(\mathbf{t})$ of path $\mathbf{t}$ is defined as the sum of the weights of its edges, i.e.,

$$W(\mathbf{t}) \triangleq \sum_{i=1}^{s} w(t_{i-1}, t_i).$$

Let us associate now to each $(M-1)$-tuple of thresholds $\mathbf{r}$, with components from the set $\mathcal{A}$, where $M \geq 1$, the M-edge path $\mathbf{t} \in \mathcal{T}(M)$, such that $r_m = a_{t_m}$ for each $1 \leq m \leq M-1$. In other words, the m-th edge on this path, which is (t_{m-1}, t_m), corresponds to the m-th magnitude cell $[r_{m-1}, r_m)$. Then it is easy to see that the weight of path $\mathbf{t}$ equals the cost $\bar{C}(\mathbf{r})$. Additionally, the above correspondence is one-to-one. Therefore, we conclude that problem (11) is equivalent to the MWP problem in the graph G, i.e., the problem of finding the path with the smallest weight, from the source to the final node.

Figure 1 illustrates the graph G (top) for the case when $K = 3$, and a path in the graph (bottom). The vertexes are represented with circles and the edges are represented with arcs. The path depicted on the bottom consists of two edges $(0, 2)$ and $(2, 4)$ and corresponds to the magnitude quantizer with bins $[0, a_2)$ and $[a_2, \infty)$.

It is known that solving the MWP problem in the WDAG G takes $O(|V| + |E|) = O(K^2)$ operations, if the edge weights can be evaluated in constant time. However, in our case, evaluating the weight of an edge requires solving the corresponding optimization problem (9). Therefore, we have to solve problem (9) for all the edges. In the next section we present an efficient way to accomplish this goal.

IV. Edge Weights Computation and Solution Algorithm

In order to be able to compute each edge weight in constant time when it is needed, we can include a preprocessing stage which solves problem (9) for all the edges and stores the

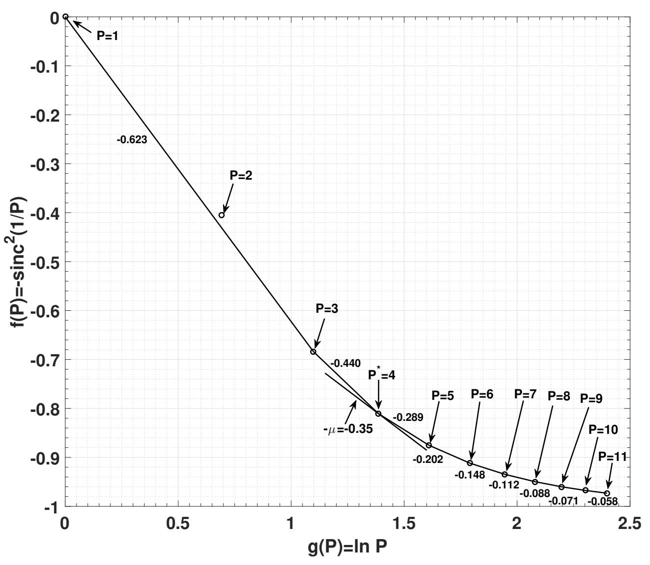


Fig. 2: Illustration of the set $\mathcal{U}$ of points of coordinates $(g(P), f(P))$, and of the set $\hat{\mathcal{U}}$, the lower boundary of the convex hull of $\mathcal{U}$, for $P_{max} = 11$. The number near each convex hull edge represents its slope. When $\mu = 0.35$ the solution to problem (14) is $P^* = 4$ since the line of slope -0.35 passing through $S(4)$ is a support line for $\mathcal{U}$. Note that $S(2)$ is the only point in $\mathcal{U}$ which is not an extreme point.

results. First we derive an important property of the optimal number of phase regions $\hat{P}(u, v)$, based on which an efficient search strategy can be developed.

Let us assume we know some value P_{max} such that

$$\hat{P}(u, v) \leq P_{max}, \text{ for all } (u, v) \in E. \tag{13}$$

We will explain later how to find such a value. Further, denote $\mathcal{P} \triangleq \{1, 2, 3, \cdots, P_{max}\}$. Moreover, for any $y > 0$, we denote $f(y) = -sinc^2(\frac{1}{y})$ and $g(y) = \ln y$ and consider the following minimization problem

$$\min_{P \in \mathcal{P}} (f(P) + \mu g(P)), \tag{14}$$

where $\mu > 0$. In view of (9) and (13) it can be easily verified that $\hat{P}(u, v)$ is a solution to problem (14) for $\mu = \frac{\lambda}{x(a_u, a_v)^2 \ln 2}$.

The straightforward approach to solve (14) is by computing the cost for each value of P and then determining the minimum. Doing so for each edge of the graph amounts to $O(K^2 P_{max})$ operations for the preprocessing step. We will show that the procedure can be considerably sped up by exploiting properties of the solutions to problem (14).

For $P \in \mathcal{P}$ let us denote by $S(P)$ the point in the plane of coordinates $(g(P), f(P))$. Additionally, let $\mathcal{U}$ denote the set of points $\{S(P) | P \in \mathcal{P}\}$. It is known [38], [39] that some value P^* minimizes the cost in (14) if and only if the point $S(P^*)$ is situated on the lower boundary of the convex hull of $\mathcal{U}$, and the line of slope $-\mu$ passing through $S(P^*)$ is a support line for $\mathcal{U}$.

Let us denote by $\hat{\mathcal{U}}$ the lower boundary of the convex hull of $\mathcal{U}$. Note that any point $S(P) \in \mathcal{U} \cap \hat{\mathcal{U}}$ is called an extreme point of $\mathcal{U}$. Consider ordering the set of extreme points in increasing order of P. Since the function $g(\cdot)$ is strictly increasing, the aforementioned order is consistent with the increasing order of $g(P)$. We will say that two extreme points are consecutive if they are consecutive with respect to the above order. Notice that since the set $\mathcal{U}$ is finite, the set $\hat{\mathcal{U}}$ is the union of line segments connecting any two consecutive extreme points. Any such line segment is called a convex hull edge. Figure 2 illustrates the sets $\mathcal{U}$ and $\hat{\mathcal{U}}$ for $P_{max} = 11$. It also shows that $P^* = 4$ is the solution to problem (14) when $\mu = 0.35$ since the line of slope -0.35 passing through $S(4)$ is a support line for $\mathcal{U}$.

Let $\hat{\mathcal{P}}$ denote the set of integers $P \in \mathcal{P}$ such that $S(P)$ is an extreme point of $\mathcal{U}$. For each $P \in \hat{\mathcal{P}}$, except for the first and the last ones, further denote by $left\_slope(P)$ (respectively, $right\_slope(P)$) the slope of the convex hull edge to the left (respectively, right) of $S(P)$, i.e., connecting $S(P)$ with the previous (respectively, next) extreme point. Then the following relations hold

$$left\_slope(P) \le right\_slope(P), \text{ for any } P \in \hat{\mathcal{P}}. \tag{15}$$

An intuitive interpretation of the above relations is that when traversing the set of convex hull edges from left to right, i.e., when moving through the extreme points in increasing order of P, the slope of the convex hull edge does not decrease. We see that condition (15) is verified in Figure 2.

Finally, the condition that the line of slope $-\mu$ passing through some extreme point $S(P)$ is a support line to $\mathcal{U}$, is equivalent to the following

$$left\_slope(P) \le -\mu \le right\_slope(P).$$

In light of the above discussion we obtain the following characterization of $\hat{P}(u,v)$, stated as a lemma.

Lemma. For each $(u,v) \in E$, the value $\hat{P}(u,v)$ equals the smallest $P \in \hat{\mathcal{P}}$ satisfying

$$left\_slope(P) \le -\frac{\lambda}{x(a_u,a_v)^2 \ln 2} \le right\_slope(P). \tag{16}$$

The above lemma together with (15) implies that $\hat{P}(u,v)$ can be found using a binary search over the set $\hat{\mathcal{P}}$. For this the knowledge of the set $\hat{\mathcal{P}}$ is needed, which is settled by the following result, proved in the appendix.

Proposition 1. $\hat{\mathcal{P}} = \mathcal{P} \setminus \{2\}$.

Note that Figure 2 confirms the above result for the case when $P_{max} = 11$.

By applying the aforementioned strategy for each graph edge leads to a time complexity of $O(K^2 \log|\hat{\mathcal{P}}|)$ for the preprocessing step. However, we will show that the complexity can be even further reduced when $P_{max} << K \log|\hat{\mathcal{P}}|$. For this we use the following monotonicity result.

Proposition 2. For any integers u, u', v, v' such that $0 \le u < v \le K+1$, $0 \le u' < v' \le K+1$, and such that $u \le u'$ and $v \le v'$, the following inequality holds:

$$\hat{P}(u,v) \le \hat{P}(u',v'). \tag{17}$$

Proof: Notice that $x(\alpha,\beta)$ is the centroid of the interval (α,β). It is known that $x(\alpha,\beta) \ge \alpha$ and that $x(\alpha,\beta)$ is a non-decreasing function of both α and β[1]. Then, under the conditions specified in the hypothesis, it follows that

$$0 \le x(a_u,a_v) \le x(a_{u'},a_{v'}),$$

which further leads to

$$-\frac{\lambda}{x(a_u,a_v)^2 \ln 2} \le -\frac{\lambda}{x(a_{u'},a_{v'})^2 \ln 2}, \tag{18}$$

since λ and $\ln 2$ are positive. The above inequality together with Lemma and relations (15) implies inequality (17), thus proving the claim. ∎

Algorithm 1: Efficient procedure to precompute all values $\hat{P}(u,v)$.

```
for $v = 1$ to $K+1$ do
    $\hat{P}(v-1,v) := \min \arg\min_{P\in\hat{\mathcal{P}}} E(P,v-1,v)$
    for $u = v-2$ down to $0$ do
        $\hat{P}(u,v) :=$
        $\min \arg\min_{\hat{P}(u,v-1)\le P\le\hat{P}(u+1,v)} E(P,u,v)$
```

Remark. Proposition 2 implies that

$$\hat{P}(u,v) \le \hat{P}(K,K+1) \text{ for all } 0 \le u < v \le K+1.$$

Then the value of $\hat{P}(K,K+1)$ can be set as P_{max}. In view of Lemma and Proposition 1, the value $\hat{P}(K,K+1)$ can be determined by inspecting all positive integers $P, P \ne 2$, in increasing order until relation (16) is satisfied.

Proposition 2 implies that the search range for $\hat{P}(u,v)$ can be reduced from $\hat{\mathcal{P}}$ to the smaller set $[\hat{P}(u,v-1),\hat{P}(u+1,v)] \cap \hat{\mathcal{P}}$, if $\hat{P}(u,v-1)$ and $\hat{P}(u+1,v)$ are evaluated first. Therefore, in order to exploit this observation we need to choose carefully the order of computation of the values $\hat{P}(u,v)$. To facilitate a visual representation of this ordering imagine that $\hat{P}(u,v)$ is the element on row u and column v of an upper triangular matrix $\hat{P}$. Note that the row indexes range from 0 to K while the column indexes range from 1 to $K+1$, hence the main diagonal contains the elements $\hat{P}(u,u+1)$, $0 \le u \le K$. We will compute the elements of this upper triangular matrix starting in the top left corner, i.e., with $\hat{P}(0,1)$, then proceeding in increasing order of the columns. Further, on each column we start with the element on the main diagonal and move up to the top.

The pseudocode of the above procedure is described in Algorithm 1, where we denote

$$E(P,u,v) \triangleq \left(-sinc^2\left(\frac{1}{P}\right) x(a_u,a_v)^2 + \lambda \log_2 P\right).$$

[1]A proof of this result can be found in [41].

In order to evaluate the running time of Algorithm 1 note that the computation of each entry on the main diagonal takes $O(P_{max})$ time, therefore $O(KP_{max})$ time is needed for all of them. On the other hand, evaluating all entries on any of the other K superdiagonals takes only $O(P_{max}+K)$ operations. To see this consider the j-th superdiagonal for some $j \geq 1$. Its elements are $\hat{P}(u, u+j+1)$, $0 \leq u \leq K-j$. The entry $\hat{P}(u, u+j+1)$ is evaluated in $O(\hat{P}(u+1, u+j+1) - \hat{P}(u, u+j)+1)$ time. Therefore, the total time for the j-th superdiagonal is

$$\begin{aligned} O&\left(\sum_{u=0}^{K-j}\left(\hat{P}(u+1,u+j+1)-\hat{P}(u,u+j)+1\right)\right) \\ &= O\left(\hat{P}(K-j+1,K+1)-\hat{P}(0,j)+K-j+1\right) \\ &= O\left(P_{max}+K\right). \end{aligned}$$

It follows that the total running time of Algorithm 1 is $O(KP_{max}+K^2)$ time, which equals $O(K^2)$ when $P_{max} < K$. Additionally, since the upper triangular matrix needs to be stored an extra $O(K^2)$ storage space is required.

In order to enable the computation of each edge weight in constant time, the following cumulative probabilities and first moments are also precomputed and stored during the preprocessing step,

$$Cum_i(u) \triangleq \int_0^{a_u} r^i g(r)dr,$$

for $i = 0, 1$, and $0 \leq u \leq K+1$, where $a_0 = 0$ and $a_{K+1} = \infty$ by convention. The values $Cum_i(u)$ can be computed in increasing order of u using

$$Cum_i(u) = Cum_i(u-1) + \int_{a_{u-1}}^{a_u} r^i g(r)dr.$$

Thus, assuming that the evaluation of each integral $\int_{a_{u-1}}^{a_u} r^i g(r)dr$ takes constant time, the computation of all these cumulative values takes $O(K)$ time. Additionally, $O(K)$ storage space is needed to store them. Based on these values, when the weight of edge (u, v) is needed, the quantities $q(a_u, a_v)$ and $x(a_u, a_v)$ will be computed in $O(1)$ time using

$$\begin{aligned} q(a_u, a_v) &= Cum_0(v) - Cum_0(u), \\ x(a_u, a_v) &= \frac{Cum_1(v) - Cum_1(u)}{q(a_u, a_v)}. \end{aligned}$$

Recall that if all values $\hat{P}(u, v)$ are precomputed, $O(K^2)$ storage space is required. If K is large and memory is an issue, we can avoid this by computing the values $\hat{P}(u, v)$ on the fly during the algorithm execution and storing them only temporarily.

This can be done by organizing the computations of the MWP algorithm such that the edges are traversed in the same order as in Algorithm 1, then computing the value $\hat{P}(u, v)$ when the edge (u, v) is traversed, and storing this value only until all the values corresponding to column $v+1$ in the upper triangular matrix $\hat{P}$ are evaluated. This way the extra memory is reduced to $O(K)$.

The following pseudocode in Algorithm 2 describes the algorithm to solve problem (7) including the above procedure for determining the values $\hat{P}(u, v)$. We point out that $\hat{W}(v)$ denotes the weight of the minimum-weight path from the source to node v, and $\varepsilon(v)$ records the node preceding v on this optimal path. At the end, the MWP can be tracked back by utilizing the values of $\varepsilon(v)$. The output is the vector $\mathbf{t}$ representing the nodes on the path. During the preprocessing stage the value of P_{max} is evaluated using $P_{max} = \hat{P}(K, K+1)$, and the cumulative probabilities and first moments are computed and stored.

Algorithm 2: Solution algorithm for problem (7).

```
Preprocessing Stage
begin
    $\hat{W}(0) = 0$
    for $v = 1$ to $K+1$ do
        Allocate memory of size $v$ to store $\hat{P}(\cdot, v)$
        $\hat{P}(v-1, v) := \min\arg\min_{P\in\hat{\mathcal{P}}} E(P, v-1, v)$
        $\hat{W}(v) := \hat{W}(v-1) + w(v-1, v)$
        $\varepsilon(v) := v-1$
        for $u = v-2$ down to $0$ do
            $\hat{P}(u, v) :=$
            $\min\arg\min_{\hat{P}(u,v-1)\leq P\leq\hat{P}(u+1,v)} E(P, u, v)$
            if $\left(\hat{W}(u) + w(u, v) < \hat{W}(v)\right)$ then
                $\hat{W}(v) := \hat{W}(u) + w(u, v)$
                $\varepsilon(v) := u$
        Deallocate memory of $\hat{P}(\cdot, v-1)$
    // Restoring the MWP using back-tracking.
    $i = K+1$
    $j = 0$
    $s(j) = i$
    while $(i \neq 0)$ do
        $j := j+1$
        $s(j) := \varepsilon(i)$
        $i := \varepsilon(i)$
    // Reverse array $s$ to obtain the vector $\mathbf{t}$.
    while $(i \leq j)$ do
        $t_i := s(j-i)$
        $i := i+1$
```

In conclusion, solving problem (7) takes $O(K^2 + KP_{max})$ time in total. If the condition $P_{max} < K$ is satisfied, which is the case in our experiments, then the total time complexity for solving problem (7) is $O(K^2)$.

V. Experimental Results

This section assesses the practical performance of the proposed ECUPQ design algorithm and compares it with the designs of [7], [14] and with ECRQ. The experiments are conducted for a two-dimensional random vector (X_1, X_2), where X_1 and X_2 are independent and identically distributed Gaussian variables with zero-mean and unit-variance. After conversion to polar coordinates the joint pdf becomes

$$p(r, \theta) = \frac{r}{2\pi}\exp\left(-\frac{r^2}{2}\right),\ 0 \leq r < \infty,\ 0 \leq \theta < 2\pi,$$

TABLE I: Performance comparison of the proposed ECUPQ with the entropy-coded UPQ of [7] and $D_G(R)$, for rates $R < 2.5$ bits/sample.

Rate	$10\log_{10} D$	$10\log_{10} D^{[7]}$	$10\log_{10} \frac{D^{[7]}}{D}$	$10\log_{10} \frac{D}{D_G(R)}$
0.500	−2.127	−1.662	0.465	0.883
0.793	−3.560	−3.344	0.216	1.211
1.000	−4.692	−4.401	0.291	1.328
1.157	−5.596	−5.100	0.496	1.369
1.278	−6.305	−5.952	0.353	1.391
1.377	−6.879	−6.517	0.362	1.411
1.570	−7.996	−7.282	0.714	1.450
1.636	−8.392	−7.721	0.671	1.460
1.754	−9.089	−8.447	0.642	1.473
1.815	−9.444	−8.762	0.682	1.479
1.948	−10.235	−9.626	0.609	1.492
2.256	−12.069	−11.314	0.755	1.515
2.422	−13.056	−12.336	0.720	1.524
2.495	−13.496	−12.774	0.722	1.527

where $r = \sqrt{x_1^2 + x_2^2}$, and $\theta = \tan^{-1}(x_2/x_1)$. It then follows that $g(r) = r\exp(-r^2/2)$.

The finite set of possible thresholds $\mathcal{A}$ is obtained by dividing the range $[0, 6]$ into subintervals of size 0.001 and picking the thresholds between intervals. In other words, $K = 6000$ and $a_i = 0.001i$, for $1 \leq i \leq K$. Moreover, we set $P_{max} = 600$ in the optimization of the number of phase regions. In order to design an ECUPQ achieving some target rate R_t we run the algorithm for various values of λ until the entropy of the UPQ becomes sufficiently close to R_t. We use D to denote the distortion (per sample) of the proposed approach, computed based on (4). The distortion is converted in dB using $10\log_{10} D$. The rate R, in bits/sample, is computed as the entropy of the ECUPQ, i.e., as $H(I_a, I_\theta)/2$.

The comparison against the entropy-coded UPQ of [7] is performed for rates in the range from 0.5 to 2.5, bits/sample based on the results reported in [7]. The comparison with the asymptotically optimal ECUPQ of [14] is performed for rates higher than 2.05.

Table I illustrates the performance comparison with [7]. Recall that the UPQ of [7] is level-constrained, i.e., it is designed with the aim of minimizing the distortion for a fixed number N of quantization bins. However, the rate reported is computed as the entropy of the quantizer. Note that all the results related to the UPQs of [7] are taken from [7]. The second last column in the table shows the gain in performance of the proposed approach versus the method of [7]. It can be seen that our algorithm always outperforms the design of [7] with gains always higher than 0.2 dB, and even larger than 0.6 dB when $R \geq 1.5$. Additionally, a peak improvement of 0.755 dB is achieved for $R = 2.256$ bits/sample.

The last column in Table I lists the gap between the ECUPQ distortion and the distortion-rate function $D_G(R)$ of a univariate Gaussian source, given by

$$D_G(R) = 2^{-2R}.$$

Note that the gap takes values between 0.883 dB, at rate $R = 0.5$, and 1.527 dB, at $R = 2.495$ bits/sample.

The vectors of thresholds $\mathbf{r}$ and the configurations $(N, M, P_1, \cdots, P_M)$ for the proposed ECUPQ and for the UPQ of [7] are presented in Table II. We observe that for the same output entropy the level-constrained UPQ of [7] has a much smaller number of quantizer bins N than our ECUPQ. The same observation holds for the number M of magnitude bins. On the other hand, such a conclusion does not hold for P_m. In particular, we see that ECUPQ has $P_1 = 1$ always, which means (since $M > 1$) that it has a disc-shaped cell around the origin (see Figure 3a), while for the UPQ of [7], P_1 can take any value between 1 and 5.

By examining the number of phase levels P_m for the proposed ECUPQ we see that for each rate, P_m increases with increasing m. This is expected in view of Proposition 2. On the other hand, it can be noticed that for ECUPQs with the same number of magnitude levels M, P_m remains the same for each $m, 1 \leq m \leq M-1$, while as M increases P_m is non-decreasing most of the time. It would be interesting to find out if the above observations can be confirmed theoretically and whether they can be exploited in order to reduce the ECUPQ design complexity. The investigation of such possibilities is deferred to future work.

Next we compare the performance of the proposed design scheme with the ECUPQ optimized in [14] based on the high resolution assumption. We will use the acronym ASY to refer to the asymptotical ECUPQ performance derived in [14]. Note that the asymptotical distortion (per sample) of ASY obtained in [14] is

$$D_{ASY} = \frac{2^{-(2R-\log_2(2\pi e))}}{12}, \tag{19}$$

for rates $R \geq 0.5\log_2(2\pi e) \approx 2.047$.

Table III illustrates the performance of the proposed algorithm in comparison with ASY for several rates in the range 2.050 to 5.996 bits/sample. We see that the proposed algorithm performs extremely close to ASY. Specifically, for the rates higher than 2.495 the absolute value of the performance difference is smaller than 0.01 dB, while for the rates lower than 2.495, our design is actually slightly better reaching improvements of up to 0.032 dB.

Table III also shows the gap between the performance of the proposed ECUPQ and the asymptotical performance (per sample) of the two-dimensional entropy-constrained vector quantizer (ECVQ), computed based on [40]

$$D_{ECVQ} = \frac{5}{36\sqrt{3}} 2^{-(2R-\log_2(2\pi e))}.$$

Moreover, the difference in performance versus the distortion-rate function is also presented in Table III. As we can observe the gap between the proposed scheme and ECVQ is small, taking values from 0.135 dB to 0.175 dB, while the gap to the theoretical limit given by the distortion-rate function, ranges from 1.501 dB to 1.541 dB.

In addition, we also compare the proposed ECUPQ design with the practical ECUPQ based on the asymptotic point density functions given in [14]. We refer to the latter scheme using the acronym PASY. The asymptotically optimal magnitude and phase quantization point densities, denoted by $g_A(a)$, $g_\Theta(\theta, a)$, respectively, which are defined as the inverse of the

TABLE II: Configuration of the proposed ECUPQ, of the entropy-coded UPQ of [7] and of the optimal ECRQ, for rates $R < 2.5$ bits/sample.

Rate	$(N, M, P_1, \cdots, P_M)$	$\mathbf{r}$	$(N, M, P_1, \cdots, P_M)^{[7]}$	$\mathbf{r}^{[7]}$	N^{ECRQ}	$\mathbf{r}^{ECRQ}$
0.500	$(7, 2, 1, 6)$	(1.947)	$(2, 1, 2)$	–	9	$(-1.730, 1.728)$
0.793	$(22, 3, 1, 6, 15)$	$(1.593, 4.892)$	$(3, 1, 3)$	–	25	$(-4.833, -1.408, 1.407, 4.832)$
1.000	$(20, 3, 1, 6, 13)$	$(1.360, 3.987)$	$(4, 1, 4)$	–	25	$(-3.979, -1.210, 1.209, 3.978)$
1.157	$(20, 3, 1, 6, 13)$	$(1.185, 3.384)$	$(5, 2, 1, 4)$	(0.752)	25	$(-3.422, -1.067, 1.066, 3.421)$
1.278	$(40, 4, 1, 6, 12, 21)$	$(1.060, 2.973, 5.437)$	$(6, 2, 1, 5)$	(0.752)	25	$(-3.038, -0.963, 0.961, 3.036)$
1.377	$(39, 4, 1, 6, 12, 20)$	$(0.971, 2.695, 4.780)$	$(7, 2, 1, 6)$	(0.752)	49	$(-5.067, -2.759, -0.884, 0.883, 2.758, 5.066)$
1.570	$(38, 4, 1, 6, 12, 19)$	$(0.827, 2.265, 3.885)$	$(9, 2, 3, 6)$	(1.066)	49	$(-4.068, -2.317, -0.759, 0.744, 2.301, 4.049)$
1.636	$(68, 5, 1, 7, 13, 19, 28)$	$(0.839, 2.272, 3.815, 5.633)$	$(10, 2, 3, 7)$	(1.051)	64	$(-4.721, -2.979, -1.453, -0.019\ 1.415, 2.937, 4.671)$
1.754	$(67, 5, 1, 7, 13, 19, 27)$	$(0.767, 2.066, 3.430, 4.952)$	$(12, 2, 4, 8)$	(1.163)	64	$(-4.161, -2.669, -1.308, -0.005\ 1.299, 2.659, 4.150)$
1.815	$(66, 5, 1, 7, 13, 19, 26)$	$(0.733, 1.971, 3.259, 4.670)$	$(13, 2, 5, 8)$	(1.247)	81	$(-4.191, -2.775, -1.473, -0.226, 1.014, 2.290, 3.656, 5.185)$
1.948	$(99, 6, 1, 7, 13, 19, 26, 33)$	$(0.664, 1.779, 2.921, 4.1345, 5.4675)$	$(16, 3, 1, 6, 9)$	$(0.475, 1.400)$	100	$(-5.211, -3.861, -2.627, -1.464, -0.338\ 0.781, 1.918, 3.104, 4.378)$
2.256	$(136, 7, 1, 7, 13, 19, 25, 32, 39)$	$(0.530, 1.414, 2.305, 3.217, 4.163, 5.157)$	$(25, 3, 4, 10, 11)$	$(0.798, 1.674)$	169	$(-5.212, -4.195, -3.227, -2.295, -1.387, -0.495, 0.394, 1.286, 2.191, 3.119, 4.082, 5.093)$
2.422	$(180, 8, 1, 7, 13, 19, 25, 32, 38, 45)$	$(0.470, 1.253, 2.040, 2.838, 3.655, 4.497, 5.368)$	$(32, 4, 1, 7, 12, 12)$	$(0.363, 1.031, 1.846)$	196	$(-4.916, -4.044, -3.204, -2.385, -1.582, -0.789, -0.001, 0.788, 1.581, 2.384, 3.203, 4.044, 4.914)$
2.495	$(180, 8, 1, 7, 13, 19, 25, 32, 38, 45)$	$(0.445, 1.188, 1.933, 2.687, 3.455, 4.243, 5.056)$	$(36, 4, 1, 8, 13, 14)$	$(0.369, 1.051, 1.848)$	256	$(-5.461, -4.628, -3.819, -3.032, -2.261, -1.502, -0.751, -0.004, 0.744, 1.495, 2.254, 3.024, 3.811, 4.620, 5.453)$

TABLE III: Performance comparison of the proposed ECUPQ with ASY, ECVQ and $D_G(R)$, for rates $R \geq 0.5\log_2(2\pi e)$ bits/sample.

Rate	$10\log_{10} D$	$10\log_{10} D_{ASY}$	$10\log_{10}\frac{D_{ASY}}{D}$	$10\log_{10}\frac{D}{D_{ECVQ}}$	$10\log_{10}\frac{D}{D_G(R)}$
2.050	−10.842	−10.810	0.032	0.135	1.501
2.151	−11.442	−11.417	0.025	0.143	1.509
2.256	−12.069	−12.051	0.018	0.150	1.515
2.422	−13.056	−13.046	0.010	0.158	1.524
2.495	−13.496	−13.490	0.006	0.161	1.527
2.998	−16.511	−16.517	−0.006	0.173	1.539
3.498	−19.517	−19.524	−0.007	0.175	1.541
4.000	−22.542	−22.550	−0.008	0.174	1.540
4.500	−25.557	−25.560	−0.003	0.172	1.538
4.995	−28.538	−28.540	−0.002	0.171	1.536
5.496	−31.555	−31.556	−0.001	0.170	1.536
5.996	−34.560	−34.564	−0.004	0.170	1.536

TABLE IV: Performance comparison of the proposed ECUPQ with PASY.

Rate	$10\log_{10} D$	$10\log_{10} D_{PASY}$	$10\log_{10}\frac{D_{PASY}}{D}$
2.050	−10.842	−10.223	0.619
2.151	−11.442	−10.841	0.601
2.256	−12.069	−11.491	0.578
2.422	−13.056	−12.521	0.535
2.495	−13.496	−12.983	0.513
2.998	−16.511	−16.154	0.357
3.498	−19.517	−19.297	0.220
4.000	−22.542	−22.418	0.124
4.500	−25.557	−25.491	0.066
4.995	−28.538	−28.504	0.034
5.496	−31.555	−31.538	0.017
5.996	−34.560	−34.553	0.007

corresponding quantization step sizes, are derived in [14] as

$$g_A(a) = \sqrt{\frac{1}{6\lambda \log_2(e)}}, \tag{20}$$

$$g_\Theta(\theta, a) = \sqrt{\frac{a^2}{6\lambda \log_2(e)}}, \tag{21}$$

where $\lambda > 0$ is the Lagrangian multiplier and a denotes the reconstructed magnitude value. Notice that the magnitude quantizer corresponding to (20) is uniform with step size $1/g_A(a)$, while the phase quantizer corresponding to (21) has step size $1/g_\Theta(\theta, a)$.

To implement the UPQ based on equations (20) and (21) we proceed as follows. The vector $\mathbf{r}$ of thresholds of the magnitude quantizer is obtained by dividing the interval $[0, 6]$ in subintervals of size $1/g_A(a)$. The middle of each magnitude quantizer bin is taken as the reconstruction, except for the first bin, for which the reconstruction is always set to 0. Subsequently, the number of phase regions corresponding to each magnitude level is computed as the value of $2\pi g_\Theta(\theta, a)$ rounded to the closest integer, where a is reconstruction value of the magnitude. Moreover, the quantized phase value is taken as the middle of the corresponding phase region as in (2). We evaluate the distortion and entropy of PASY using (1), respectively, $H(I_a, I_\theta)/2$, where $H(I_a, I_\theta)$ is given in (5).

Table IV depicts the performance of the proposed ECUPQ in comparison with PASY for rates from 2.050 to 5.996 bits/sample. It can be observed that the proposed algorithm outperforms PASY for all rates examined. The performance improvement is between 0.5 and 0.619 dB for rates up to 2.495. The gap gradually decreases as the rate increases, but it still remains higher than 0.1 dB for rates up to 4. Finally, for $R \approx 5.996$ the gap falls below 0.01 dB.

TABLE V: Performance comparison of the proposed ECUPQ against ECRQ.

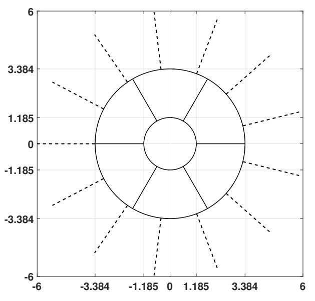
Rate	$10\log_{10} D$	$10\log_{10} D_{ECRQ}$	$10\log_{10} \frac{D_{ECRQ}}{D}$
0.500	−2.127	−2.093	0.034
0.793	−3.560	−3.483	0.077
1.000	−4.692	−4.579	0.113
1.157	−5.596	−5.470	0.126
1.278	−6.305	−6.180	0.125
1.377	−6.879	−6.767	0.112
1.570	−7.996	−7.920	0.076
1.636	−8.392	−8.321	0.071
1.754	−9.089	−9.030	0.059
1.815	−9.444	−9.393	0.051
1.948	−10.235	−10.192	0.043
2.256	−12.069	−12.053	0.016
2.422	−13.056	−13.048	0.008

Since the authors of [14] show that the asymptotical performance of ECUPQ and of ECRQ are identical, we are interested in comparing the proposed approach against ECRQ at small rates. For this we implement the ECRQ using as the scalar quantizer for each Cartesian coordinate the entropy-constraint scalar quantizer designed using the algorithm of [28]. We point out that the algorithm of [28] guarantees the globally optimal solution for the problem of minimizing the Lagrangian, when the quantizer thresholds are confined to a finite set. For fairness of comparison we use the same discretization step size as for ECUPQ. In other words, to obtain the finite set of possible thresholds we divide the interval $[-6, 6]$ in subintervals of size 0.001. This algorithm was used to optimize the ECRQ for rates in the range from 0.5 to 6 bits/sample.

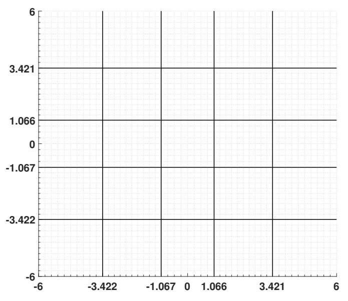
We list in Table II the number of quantization levels N^{ECRQ} of the optimal ECRQ, and the corresponding vector of thresholds $\mathbf{r}^{ECRQ} = \{r_1^{ECRQ}, r_2^{ECRQ}, \cdots, r_{\sqrt{N'}-1}^{ECRQ}\}$ for the scalar quantizer partition, where $r_0^{ECRQ} = -\infty$ and $r_{\sqrt{N'}}^{ECRQ} = \infty$ by default, for various rates between 0.5 and 2.495 bits/sample. The performance comparison between ECUPQ and ECRQ is illustrated in Table V only for the range of rates from 0.5 to 2.422, since for higher rates the absolute value of the performance difference is less than 0.01 dB. The results in Table V show that the proposed ECUPQ outperforms ECRQ in the low-rate region with improvements reaching up to 0.126 dB. Specifically, the performance improvement first increases as the rate increases up to about 1.2 bits/sample, after which it gradually deceases. We note that the gap remains above 0.1 dB for rates between 1 and 1.377.

In order to understand why ECUPQ is better than ECRQ at low rates it is instructive to analyze the structure of the quantizer partition. This is depicted in Figure 3a for the ECUPQ at rate 1.157 and in Figure 3b for the ECRQ at the same rate. Two possible reasons for the superiority of ECUPQ at low rates are:

1) ECUPQ has higher flexibility in the choice of the number N of quantization bins, as it can be seen in Table II. Namely, for ECUPQ N could be any positive integer, while for ECRQ N^{ECRQ} can only be a perfect square. We see that the ECUPQ with rate 1.157 has $N = 20$, while the ECRQ cannot select such a value for N^{ECRQ}. Instead it has $N^{ECRQ} = 25$. The same conclusion holds for all the rates illustrated in Table II.
2) We observe in Figure 3a that the cell in the center of the ECUPQ is a disc, which is the perfect shape to minimize the distortion, while in ECRQ all cells are squares. Actually, by inspecting the configurations $(N, M, P_1, \cdots, P_M)$ in Table II, we see that the proposed ECUPQ always has a disc-shaped cell around the origin.



(a) ECUPQ.



(b) ECRQ.

Fig. 3: The partitions of proposed ECUPQ (a) and ECRQ (b) at rate $R = 1.157$ bits/sample.

To summarize, we conclude that the proposed ECUPQ design algorithm outperforms the algorithm of [7] and PASY at low rates, reaching peak improvements of 0.755 dB and 0.619 dB, respectively. We point out that the peak improvements are achieved for rates lower than 2.495 bits/sample. Additionally, the proposed scheme is slightly better than ECRQ for rates $R \leq 2.256$, with improvements of up to 0.126 dB achieved at $R = 1.157$ bits/sample. Additionally, for rates higher than 2.050 our ECUPQ is extremely close in performance to ASY and is only about 0.175 dB away from the asymptotic ECVQ, while maintaining a lower implementation complexity.

Before ending this section we would like to briefly address the problem of choosing the set $\mathcal{A}$ of possible thresholds. Although the focus of this work is on solving problem (7), which assumes that the set $\mathcal{A}$ is given, the choice of $\mathcal{A}$ determines how well the solution of (7) approximates the solution to the unconstrained problem (6). A straightforward choice for the set $\mathcal{A}$ is the one we used in our experiments, namely, to divide some interval $[0, B]$ into small intervals of equal size Δ. The smaller the value of Δ, the better the approximation. However, if the function $g(r)$ is decreasing for r larger than some value $r_0 \in [0, B]$, it is natural to think that an error of size Δ at a higher magnitude quantizer threshold has a smaller impact on the performance than an error of the same size at a smaller threshold. This suggests that a non-uniform discretization of the interval $[0, B]$, where the sizes of the sub-intervals start with Δ, but gradually increase, may lead to the same performance, but with reduced time and space complexities since the value of K would be lower. The investigation of such a possibility deserves attention and we will address it in future work.

VI. Conclusion

This paper focuses on the design of entropy-constrained unrestricted polar quantizer for bivariate circularly symmetric sources. We propose a design algorithm which is globally optimal when the thresholds of the magnitude quantizer are confined to a finite set. Our solution algorithm consists of solving the minimum-weight path problem in a certain weighted directed acyclic graph, in conjunction with an efficient procedure to find the optimal number of phase regions for each possible magnitude quantizer bin. The experimental results, performed for a bivariate circularly symmetric Gaussian source, demonstrate significant improvements over the prior practical designs at rates up to 2.5 bits/sample, and performance very close to the optimal asymptotical performance.

References

[1] G. H. Senge, "Quantization of image transforms with minimum distortion", *Technical Report No. ECE-77-8,* Dept. of Elec. and Comp. Eng., University of Wisconsin, Madison, WI, Jun. 1997.

[2] N. C. Gallagher, Jr., "Quantizing schemes for the discrete Fourier transform of a random time-series," *IEEE Trans. Inform. Theory*, vol. IT-24, no. 2, pp. 156-163, Mar. 1978.

[3] W. A. Pearlman and R. M. Gray, "Source coding of the discrete Fourier transform", *IEEE Trans. Inform. Theory,* vol. IT-24, no. 6, pp. 683-692, Nov. 1978.

[4] W. A. Pearlman, "Polar quantization of a complex Gaussian random variable", *IEEE Trans. Commun.,* vol. COM-27, no. 6, pp. 892-899, Jun. 1979.

[5] J. A. Bucklew and N. C. Gallagher, Jr., "Quantization schemes for bivariate Gaussian random variables," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 537-543, Sept. 1979.

[6] J. A. Bucklew and N. C. Gallagher, Jr., "Two-dimensional quantization of bivariate circularly symmetric densities," *IEEE Trans. Inform. Theory*, vol. IT-25, no. 5, pp. 667-671, Nov. 1979.

[7] S. G. Wilson, "Magnitude/phase quantization of independent Gaussian variates", *IEEE Trans. Commun.,* vol. COM-28, no. 11, pp. 1924-1929, Nov. 1980.

[8] P. F. Swaszek and J. B. Thomas, "Optimal circularly symmetric quantizers," *Franklin Inst. J.*, vol. 313, no. 6, pp. 373-384, Jun. 1982.

[9] P. F. Swaszek, "Uniform spherical coordinate quantization of spherically symmetric sources," *IEEE Trans. Commun.*, vol. COM-33, no. 6, pp. 518-521, Jun. 1985.

[10] P. F. Swaszek and T.W. Ku, "Asymptotic performance of unrestricted polar quantizers," *IEEE Trans. Inform. Theory,* vol. IT-32, no. 2, pp. 330-333, Mar. 1986.

[11] D. L. Neuhoff, "Polar quantization revisited," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT 1997),* pp. 60, Ulm, Germany, Jun. 1997.

[12] P. W. Moo and D. L. Neuhoff, "Uniform Polar Quantization Revisited", in *Proc. IEEE Int. Symp. Inform. Theory (ISIT 1998),* pp. 100, Cambridge, MA, USA, Aug. 1998.

[13] Z. H. Peric and M. C. Stefanovic, "Asymptotic analysis of optimal uniform polar quantization," *AEU Int. J. Electron. Commun.,* vol. 56, no. 5, pp. 345-347, 2002.

[14] R. Vafin and W. B. Kleijn, "Entropy-constrained polar quantization and its application to audio coding", *IEEE Trans. Speech and Audio Process.,* vol. 13, no. 2, pp. 220-232, Mar. 2005.

[15] M. D. Petkovic, Z. H. Peric, and A. Z. Jovanovic, "An iterative method for optimal resolution-constrained polar quantizer design," *COMPEL: Int. J. Comput. and Math. Elect. and Electron. Eng.,* vol. 30, no. 2, pp. 574-589, 2011.

[16] Z. Peric and J. Nikolic, "Design of asymptotically optimal unrestricted polar quantizer for Gaussian source", *IEEE Signal Process. Lett.,* vol. 20, no. 10, pp. 980-983, Oct. 2013.

[17] A. Z. Jovanovic, Z. H. Peric, J. R. Nikolic and M. R. Dincic, "Asymptotic analysis and design of restricted uniform polar quantizer for Gaussian sources", *Digital Signal Process.,* vol. 49, pp. 24-32, Feb. 2016.

[18] H. Pobloth, R. Vafin, and W. B. Kleijn, "Multivariate block polar quantization", *IEEE Trans. Commun.,* vol. 53, no. 12, pp. 2043-2053, Dec. 2005.

[19] E. Ravelli and L. Daudet, "Embedded polar quantization", *IEEE Signal Process. Lett.,* vol. 14, no. 10, pp. 657-660, Oct. 2007.

[20] N. Kingsbury and T. Reeves, "Redundant representation with complex wavelets: How to achieve sparsity", in *Proc. Int. Conf. Image Process.(ICIP 2003),* pp. 45-48, Barcelona, Spain, Sep. 2003.

[21] A. M. Bruckstein, R. J. Holt and A. N. Netravali, "Holographic representations of images", *IEEE Trans. Image Process.,* vol. 7, no. 11, pp. 1583-1597, Nov. 1998.

[22] P. Nazari, B-K. Chun, F. Tzeng and P. Heydari, "Polar quantizer for wireless receivers: theory, analysis, and CMOS implementation", *IEEE Trans. Circuits and Systems-I: Regular Papers,* vol. 61, no. 3, pp. 877-887, Mar. 2014.

[23] J. Max, "Quantizing for minimum distortion", *IRE Trans. Inform. Theory,* vol. 6, no. 1, pp. 7-12, Mar. 1960.

[24] S. P. Lloyd, "Least squares quantization in PCM", *IEEE Trans. Inform. Theory,* vol. IT-28, no. 2, pp. 129-137, Mar. 1982.

[25] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Magazine,* vol. 18, no. 5, pp. 36-58, Sep. 2001.

[26] A. Aggarwal, B. Schieber, and T. Tokuyama, "Finding a minimum weight k-link path in graphs with the concave monge property and applications," *Discr. Computat. Geometry,* vol. 12, no. 1, pp. 263-280, Dec. 1994.

[27] D. Muresan and M. Effros, "Quantization as histogram segmentation: globally optimal scalar quantizer design in network systems", in *Proc. Data Compression Conf. (DCC 2002)*, pp. 302-311, Snowbird, UT, USA, Apr. 2002.

[28] D. Muresan and M. Effros, "Quantization as Histogram Segmentation: Optimal Scalar Quantizer Design in Network Systems," *IEEE Trans. Inform. Theory,* vol. 54, no. 1, pp. 344-366, Jan. 2008.

[29] S. Dumitrescu and X. Wu, "Optimal Multiresolution Quantization for Scalable Multimedia Coding," in *Proc. IEEE Inform. Theory Workshop (ITW 2002),* pp. 139-142, Bangalore, India, Oct. 2002.

[30] S. Dumitrescu and X. Wu, "Optimal two-description scalar quantizer design," *Algorithmica,* vol. 41, no. 4, pp. 269-287, Feb. 2005.

[31] S. Dumitrescu and X. Wu, "Lagrangian optimization of two-description scalar quantizers," *IEEE Trans. Inform. Theory,* vol. 53, no. 11, pp. 3990-4012, Nov. 2007.

[32] S. Dumitrescu, "On the Design of Optimal Noisy Channel Scalar Quantizer with Random Index Assignment," *IEEE Trans. Inform. Theory,* vol. 62, no. 2, pp. 724-735, Feb. 2016.

[33] H. Wu and S. Dumitrescu, "Design of optimal entropy-constrained scalar quantizer for sequential coding of correlated sources," *to appear* in *Proc. IEEE Inform. Theory Workshop (ITW 2017)*, Kaohsiung, Taiwan, Nov. 2017.

[34] X. Wu, "Optimal quantization by matrix searching," *J. Algorithms*, vol. 12, no. 4, pp. 663-673, Dec. 1991.

[35] H. Wu and S. Dumitrescu,"Design of optimal entropy-constrained unrestricted polar quantizer for bivariate circularly symmetric sources", *submitted to IEEE Int. Conf. Acoust., Speech, and Signal Process. (ICASSP), 2018.*

[36] P. Vogel, "Source coding by classification-91-715," *IEEE Trans. Commun.,* vol. 43, no. 11, pp. 2821-2832, Nov. 1995.

[37] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization", *IEEE Trans. Acoust., Speech, Signal Process.,* vol. 37, no. 1, pp. 31-42, Jan. 1989.

[38] H. Everett III, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Operat. Res.,* vol. 11, no. 3, pp. 399-417, Jun. 1963.

[39] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1969.

[40] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory,* vol. IT-25, no. 4, pp. 373-380, Jul. 1979.

[41] A. V. Trushkin, "Sufficient conditions for uniqueness of a locally optimal quantizer for a class of convex error weighting functions", *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 187-198, Mar. 1982.

Appendix

In this appendix we present the proof of Proposition 1.

Proof of Proposition 1: Recall that $f(P) = -sinc^2(\frac{1}{P})$, and $g(P) = \ln P$. Let us make the change of variable $u = g(P)$, for $P \geq 1$. Then $P = g^{-1}(u) = e^u$, for $u \geq 0$. Further, define

$$y(u) \triangleq f(g^{-1}(u)) = -sinc^2(e^{-u}),\ \text{for } u \geq 0.$$

It follows that the following equality holds $\mathcal{U} = \{(u, y(u)) | u \in \{\ln 1, \ln 2, \cdots, \ln P_{max}\}\}$. Thus, in order to find the lower convex hull of $\mathcal{U}$, it is useful to determine the intervals on which the function $y(u)$ is convex, when u takes values in the continuous domain $(0, \infty)$.

By computing the second order derivative of $y(u)$, we obtain

$$y''(u) = \frac{1}{x(u)^2} \underbrace{\left(-2 + (2 - 2x(u)^2)\cos(2x(u)) + 3x(u)\sin(2x(u))\right)}_{\beta(x(u))},$$

where $x(u) = \pi e^{-u}$. Note that when $u \geq 0$ we have $x(u) \in (0, \pi]$. Further, we aim at determining the sign of $\beta(x)$ as a function of x instead of u, for $x \in (0, \pi]$. For this we compute the first and second order derivatives

$$\beta'(x) = 2x\cos 2x - \sin 2x + 4x^2 \sin 2x,$$
$$\beta''(x) = 4x \underbrace{(\sin 2x + 2x\cos 2x)}_{\gamma(x)}.$$

Next we will determine the sign of $\gamma(x)$. For this divide first the domain of x into the following intervals: $I_1 = (0, \pi/4]$, $I_2 = (\pi/4, \pi/2]$, $I_3 = (\pi/2, 3\pi/4]$ and $I_4 = (3\pi/4, \pi]$. Note that when $x \in I_1$, we have $x > 0$, $\sin 2x > 0$ and $\cos 2x \geq 0$, which lead to $\gamma(x) > 0$. Additionally, for $x \in I_3$ we have $x > 0$, $\sin 2x < 0$ and $\cos 2x \leq 0$, yielding $\gamma(x) < 0$. Further, to determine the sign of $\gamma(x)$ on I_2 and I_4 we will analyze its derivative

$$\gamma'(x) = 4(\cos 2x - x\sin 2x).$$

It can be easily seen that $\gamma'(x) < 0$ holds for $x \in I_2$, while $\gamma'(x) > 0$ holds for $x \in I_4$. These imply that $\gamma(x)$ is decreasing for $x \in I_2$ and increasing for $x \in I_4$.

Further, we obtain that for $x \in I_2$, $\gamma(x)$ decreases from $\gamma(\pi/4) = 1 > 0$ to $\gamma(\pi/2) = -\pi < 0$, yielding that there exists a unique point $x_1 \in I_2$ where γ changes signs from positive to negative. In other words, $\gamma(x_1) = 0$, $\gamma(x) > 0$ for $x \in (\pi/4, x_1)$ and $\gamma(x) < 0$ for $x \in (x_1, \pi/2]$.

Similarly, for $x \in I_4$, we have that $\gamma(x)$ increases from $\gamma(3\pi/4) = -1 < 0$ to $\gamma(\pi) = 2\pi > 0$, which implies that there exists a unique point $x_2 \in I_4$ where γ changes signs from negative to positive. In other words, $\gamma(x_2) = 0$, $\gamma(x) < 0$ for $x \in (3\pi/4, x_2)$ and $\gamma(x) > 0$ for $x \in (x_2, \pi]$.

By summarizing the analysis of the sign of $\gamma(x)$ and using the fact that $\beta''(x)$ has the same sign as $\gamma(x)$, we conclude that $\beta''(x) > 0$ holds for $(0, x_1)$ and $(x_2, \pi]$, while $\beta''(x) < 0$ holds for $x \in (x_1, x_2)$. This observation implies that: 1) $\beta'(x)$ increases for $x \in [0, x_1]$ from $\beta'(0) = 0$ to $\beta'(x_1)$ (thus $\beta'(x_1)$ must be positive); 2) $\beta'(x)$ decreases for $x \in [x_1, x_2]$; 3) $\beta'(x)$ increases again for $x \in [x_2, \pi]$ up to $\beta'(\pi) = 2\pi > 0$. Using further the fact that $\beta'(\pi/2) = -\pi < 0$, we conclude that there are exactly two points $x_3, x_4 \in (0, \pi]$, $x_3 < x_4$, where $\beta'(x)$ changes signs. Specifically, we have $\beta'(x_3) = \beta'(x_4) = 0$, $\beta'(x) > 0$ for $x \in (0, x_3) \cup (x_4, \pi]$, and $\beta'(x) < 0$ for $x \in (x_3, x_4)$.

The aforementioned observation implies that $\beta(x)$ increases on $(0, x_3]$ (from $\beta(0) = 0$ to a value which must be positive), further, $\beta(x)$ decreases on $[x_3, x_4]$ and increases again on $[x_4, \pi]$ up to $\beta(\pi) = -2\pi^2 < 0$. Considering the fact that $\beta(\pi/2) = \frac{\pi^2}{2} - 4 > 0$, it follows that there exists a unique point $x_5 \in (\pi/2, \pi)$ such that $\beta(x_5) = 0$, $\beta(x) > 0$ holds for $x \in (0, x_5)$ and $\beta(x) < 0$ for $x \in (x_5, \pi]$.

Let u_0 be the unique point in $(0, \infty)$ such that $x(u_0) = x_5$. Since the sign of $y''(u)$ coincides with the sign of $\beta(x(u))$ we conclude that $y(u)$ is concave for $u \in [0, u_0)$ and is convex for $u \in [u_0, \infty)$. Further, the fact that $x_5 > \pi/2$ implies that $u_0 < \ln 2$. Leading to the conclusion that $y(u)$ is convex on $[\ln 2, \infty)$.

Recall that $S(P)$ denotes the point in the plane of coordinates $(g(P), f(P))$. Then the above considerations imply that the elements of $\hat{\mathcal{P}}$ are 1 and all points $P_0 + i$, for $0 \leq i \leq P_{max} - P_0$, where P_0 is the smallest integer larger than $\frac{\pi}{x_5}$, such that the slope of segment $(S(0), S(P_0))$ is smaller than or equal to the slope of segment $(S(P_0), S(P_0{+}1))$. We found numerically that $P_0 = 3$, thus the conclusion follows.

■

PLACE
PHOTO
HERE

Huihui Wu (S'14) received the B.Sc degree from Southwest University for Nationalities, Chengdu, China, in 2011 and the M.S. degree from Xiamen University, Xiamen, China, in 2014. Both are in communication engineering. He is currently pursuing the Ph.D. degree in electrical and computer engineering at McMaster University, Hamilton, Canada. His research interests include channel coding, joint source and channel coding, multiple description coding and signal quantization.

PLACE
PHOTO
HERE

Sorina Dumitrescu (M'05-SM'13) received the B.Sc. and Ph.D. degrees in mathematics from the University of Bucharest, Romania, in 1990 and 1997, respectively. From 2000 to 2002 she was a Postdoctoral Fellow in the Department of Computer Science at the University of Western Ontario, London, Canada. Since 2002 she has been with the Department of Electrical and Computer Engineering at McMaster University, Hamilton, Canada, where she held Postdoctoral, Research Associate, and Assistant Professor positions, and where she is currently an Associate Professor. Her current research interests include multimedia coding and communications, network-aware data compression, multiple description codes, joint source-channel coding, signal quantization. Her earlier research interests were in formal languages and automata theory. Dr. Dumitrescu held an NSERC University Faculty Award during 2007-2012.